Commodore
COMPUTER
CLUB
46
L. 4.000
La rivista degli utenti di sistemi Commodore
25 ottobre 1987 - Anno VI - N° 46 - Sped. Abb. Post. Gr. III/70 - CR - Distr. MePe
Speciale: la
multi-programmazione
MS-DOS
& GW Basic
emulator 2
Signora Amiga,
ai vostri comandi
Pronostici
Totocalcio
Auto Run per C128
Test di coppia
systems

Sommario

INSERTO

PRINCIPI DI MULTI-PROGRAMMAZIONE

RUBRICHE

Direttore: Alessandro de Simone - **Caporedattore:** Michele Maggi
Redazione/collaboratori: Paolo Agostini, Claudio Balocchi, Carlo e Lorenzo Barazzetta, Simone Bettola, Luigi Callegari, Sergio Camici, Sandro Certi, Umberto Colapicchioni, Maurizio Dell'Abate, Valerio Ferri, Giancarlo Mariani, Roberto Marigo, Clizio Merli, Marco Miotti, Roberto Morassi, Antonio Pastorelli, Carla Rampi, Marco Saetta, Fabio Sorgato, Danilo Toma, Giovanni Valli.
Segretaria di redazione: Maura Ceccaroli **Grafica:** Arturo Ciaglia, Gabriella Galbusera
Copertina: La foto di copertina è stata gentilmente concessa da Canale 5
Direzione, redazione, pubblicità: v.le Famagosta, 75 - 20142 Milano - Tel. 02/8467348
Pubblicità: Milano: Leandro Nencioni (direttore vendite), Guido Agosti, Giorgio Ruffoni, Claudio Tidone - v.le Famagosta, 75 - 20142 Milano - Tel. 02/8467348
• Emilia Romagna: Spazio E - P.zza Roosevelt, 4 - 40123 Bologna - Tel. 051/236979
• Toscana, Marche, Umbria: Mercurio srl - via Rodari, 9 - San Giovanni Valdarno (Ar) - Tel. 055/947444
• Lazio, Campania: Spazio Nuovo - via P. Foscari, 70 - 00139 Roma - Tel. 06/8109679
Segreteria: Marina Vantini - **Abbonamenti:** Liliana Spina
Tariffe: prezzo per copia L. 4.000. Abbonamento annuo (11 fascicoli) L. 40.000. Estero: il doppio. Abbonamento cumulativo alle riviste Computer e Commodore Computer Club L. 80.000. I versamenti vanno indirizzati a: Systems Editoriale Srl mediante assegno bancario o utilizzando il c/c postale n. 37952207
Composizione: Systems Editoriale Srl - **Fotolito:** Systems Editoriale Srl
Stampa: Systems Editoriale/La Litografica Srl - Busto Arsizio (Va)
Registrazioni: Tribunale di Milano n. 370 del 2/10/82 - Direttore Responsabile: Michele Di Pisa
Sped. in abb. post. gr. III - Pubblicità inferiore al 70% - **Distrib.:** MePe - via G. Carcano, 32 - Milano

Le 1000 e una Commodore

Le recenti notizie dal mondo Commodore sembrano ribadire la decisione di migliorare l'immagine dell'azienda

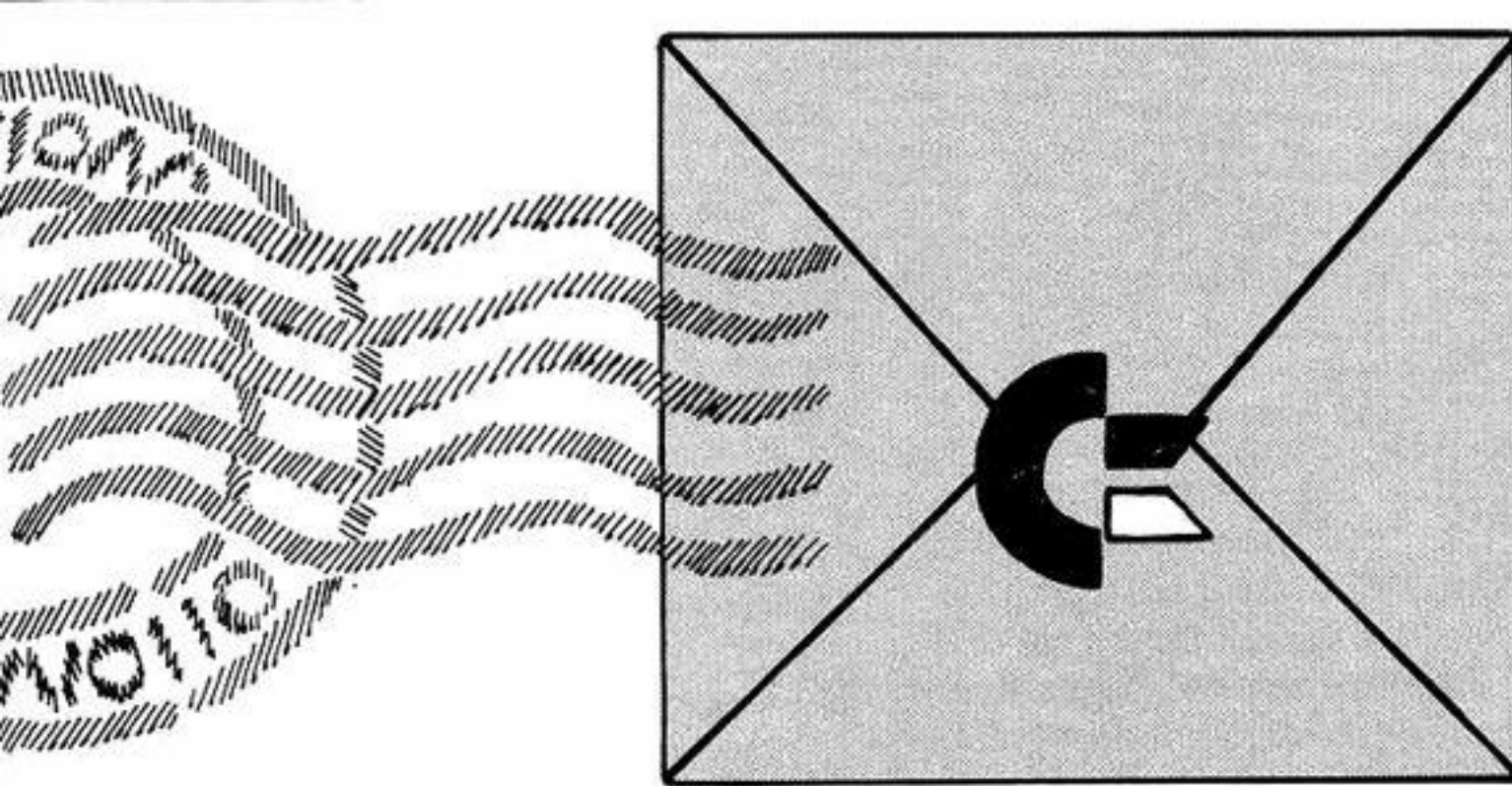
Uno degli aspetti più importanti del "nuovo corso" della Commodore, è la politica di riassetto della struttura distributiva, il cui scopo è quello di arrivare ad un contatto diretto con l'utente finale.

Per ottenere tali ambiziosi risultati, verranno identificati, attraverso una rigorosa selezione tra gli attuali negozi trattanti prodotti Commodore, 1000 esercizi commerciali che, rispondendo ai requisiti richiesti, verranno qualificati come "Commodore Point".

L'allestimento standardizzato in tutt'Italia delle loro vetrine diventerà, entro breve tempo, un punto di riferimento preciso e facilmente individuabile dal consumatore. Questi potrà disporre dei numerosi servizi offerti tra cui, fondamentale, la preparazione tecnica di personale qualificato, appositamente istruito da corsi specifici che verranno tenuti dalla stessa Commodore.

I Commodore Point, naturalmente, avranno un rapporto diretto e privilegiato con la casa-madre che preparerà, solo per loro, esclusive confezioni di prodotti informatici e garantirà, in anteprima, la presentazione di tutte le novità Commodore.

Solo presso i Commodore Point, inoltre, saranno disponibili eventuali agevolazioni promozionali che dovessero essere stabilite, e sarà possibile utilizzare gli stessi esercizi commerciali come centri di raccolta per macchine da riparare sia sotto che fuori garanzia.

La Commodore ritiene che, in un mercato così competitivo come quello della microinformatica, avere dei centri specializzati di vendita sia il giusto mezzo per seguire, in maniera costruttiva, l'evolversi dei bisogni del consumatore; soprattutto in un momento come questo, in cui l'informatica entra nelle case e negli uffici per risolvere i problemi di tutti i giorni.

la vostra posta

Colori in alta risoluzione

□ Quando cerco di alterare i colori di un disegno tracciato in alta risoluzione, a volte tutto funziona a dovere, altre volte l'operazione sembra "disturbare" piccole aree adiacenti a quella maipolata, con la creazione di effetti indesiderati: creazione di scalette colorate, modifica dei colori di altre linee adiacenti e così via. Quale può essere la causa?
(Massimiliano Di Massimo - Somewere)

• Non preoccuparti: è tutto regolare. Purtroppo i computer domestici (C/64, C/128, C/16) hanno una gestione della grafica un po' particolare che non può essere modificata.

Nonostante l'argomento sia complesso, cercherò di riassumere brevemente la tecnica adoperata per colorare lo schermo in alta risoluzione.

Dunque, iniziamo col ricordare che l'area video che si osserva quando si "entra" in alta risoluzione è formata da 64000 puntini che, a loro volta, sono l'immagine di 64000 bit che, a loro volta, sono raggruppati in gruppi di 8 in 8000 byte adiacenti (la cosiddetta "Pagina grafica").

E fin qui tutto dovrebbe esser chiaro (e noto).

Il colore di ciascun puntino, però, pur potendo variare a piacimento nella gamma dei colori disponibili (16) è rigorosamente "legato" alla sorte di altri puntini ad esso adiacenti, e vedremo subito il perchè.

Se, infatti, si volesse dotare ciascuno dei 64000 puntini della possibilità di venir colorato con uno qualunque dei 16 colori possibili, si dovrebbe rendere disponibile un'ulteriore area di memoria Ram che rappresenti, per ciascun puntino, il colore.

Siccome i colori sono 16, si renderebbero indispensabili (vedi calcolo binario) 4 bit per la rappresentazione di ciascun puntino per un totale di 16x64000= 1024000 bit che, pur se raggruppati in byte lunghi 8 bit, richiederebbero tuttavia ben 128000 byte; e questo, bada bene, solo per memorizzare le informazioni relative al colore!

Per evitare la spesa di un'area così vasta di Ram (non dimenticare l'epoca in cui il C/64 fu progettato e realizzato) i progettisti della Commodore decisero di ricorrere ad una drastica limitazione assegnando, per la memoria colore, solo 1000 (mille) byte.

Il trucco si basava sulla considerazione che, nel modo testo, una cella video destinata ad ospitare un carattere alfanumerico, è formata da una griglia di 8x8 puntini elementari ed è relativa ad un byte che, come tutti i byte, può contenere un valore compreso tra zero e 255.

Orbene, è sicuramente possibile suddividere l'area del video in alta risoluzione in mille (=25 righe x 40 colonne) mini-aree che, in modo testo, corrisponderebbero, ciascuna, ad un carattere visualizzabile.

Ogni cella che, in modo testo, può ospitare un carattere alfanumerico, rappresenta invece, in modo alta risoluzione, il colore della stessa porzione secondo un codice particolare: quattro dei suoi otto bit rappresentano il colore dei pixel accesi (posti ad 1), gli altri quattro, invece, il colore dei pixel spenti (posti a zero).

Tali codici, però, sono relativi all'intera area di 8x8 puntini che dipendono dalla locazione colore interessata.

Ne consegue, dunque, che in una stessa cella video in alta risoluzione, che in modo testo conterrebbe un carattere, puoi disporre di due soli colori che possono benissimo esser diversi dai due colori disponibili nelle celle adiacenti.

I bellissimi disegni multicolori che spesso ammiri in schermate grafiche in alta risoluzione, vengono effettuati tenendo conto di tale particolarità e manipolando con perizia la modifica dei colori passando da una cella all'altra.

Non proseguo il discorso sulla tecnica multicolor (che porta i colori a quattro, ma dimezza la risoluzione) ma ne approfitto per ricordare che con computer modesti è possibile realizzare "cose" modeste.

Per un vero CAD devi, per lo meno, passare all'Amiga.

Fast inattivo

□ Ho digitato i programmi grafici Quark e Random Project pubblicati sul N.43 (a proposito: voglio congratularmi con i bravissimi autori) ma il mio C/128 si blocca se è presente, nella riga 320, l'istruzione Fast, mentre funziona perfettamente se la elimino. Come mai?
(Adriano Locci - M. di Campo)

• Grazie per i complimenti (che giro pari pari agli autori Mauro e Stefano Ciurli di Roma) e passiamo al quesito.

Il comando Fast costringe il C/128 ad elaborare in modo più veloce grazie alla diversa frequenza con cui viene eccitato il quarzo. Per far questo, purtroppo, è necessario un piccolo sacrificio: bisogna rinunciare alla gestione del video che, non appena si attiva il comando, scompare, o meglio viene "spento" finchè non viene incontrato il comando Slow che ripristina le originarie condizioni di velocità.

Morale: benchè il comando Fast incrementi la velocità di elaborazione, è necessario dar tempo al computer affinchè la compia per intero. E' quindi probabile che la tua impazienza di osservare i risultati sia tale da ritenere difettoso il comando; oppure che hai dimenticato di digitare il comando Slow (riga 530) che ti permette di osservare i disegni dopo che sono stati tracciati.

Mappa memoria 80 colonne

□ In quale zona di memoria si trova la pagina video dedicata ai caratteri nel C/128 in modo 80 colonne?
(Christian Capone - Torvajanica)

• La gestione del video del C/128, sia in bassa che alta risoluzione, non ricalca le orme dei computer più piccoli, e l'operazione di visualizzazione non si accontenta di semplici comandi Poke.

In certi casi, addirittura, è necessario attivare via software il chip video e passargli i parametri opportuni per ottenere la conseguente visualizzazione di immagini.

Purtroppo il modesto numero di richieste relative alla gestione nel modo 80 colonne, finora pervenute in Redazione, mi impedisce di assegnare ad uno dei collaboratori il compito di scrivere un articolo specifico.

A meno che non giunga una valanga di lettere di incitamento...

Esperienze

□ Che cos'è un sistema esperto?
(Matteo Baccan - Cannobio, No)

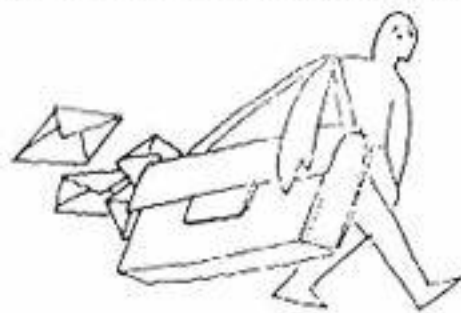
• La possibilità offerta dai moderni computer di elaborare, senza affaticarsi, milioni di operazioni, confronti e verifiche, ha spinto gli informatici a sviluppare quella che, da qualche decennio, viene definita "Intelligenza artificiale" (abbreviata: AI) e che ora tendono a chiamare, meno presuntuosamente, "Tecnologia della conoscenza".

Un sistema esperto consiste nell'assegnare una certa quantità di informazioni, catalogabili nel modo più disparato, e rintracciare, in seguito, eventuali legami che le accomunano. Naturalmente il problema più importante è proprio il modo di insegnare al computer ad effettuare le ricerche e, soprattutto, a memorizzare le varie procedure via via eseguite in modo da far tesoro dell'esperienza accumulata ed evitare eventuali errori già commessi.

Un esempio di sistema esperto (in scala molto ridotta) è quello, pubblicato più volte sulle riviste del settore, in cui si insegna al calcolatore a riconoscere un animale in base alla descrizione "indiretta" (mediante risposte "sì" e "no") che l'utente ha compiuto in precedenti colloqui con il computer. Supponiamo che il computer abbia in memoria le caratteristiche di una decina di animali e che, tra questi, sia assente il gatto. Indicando con "U" l'utente e "C" il calcolatore, ecco un esempio di colloquio che viene svolto tra "U" e "C":

C: E' un quadrupede?
U: sì
C: Vive anche in acqua?
U: no
C: E' erbivoro?
U: no
C: E' un canide?
U: no
C: è un felino?
U: sì
C: Ha la criniera?
U: no
C: E' la tigre?
U: no
C: Come si chiama l'animale?
U: Gatto
C: in che cosa il GATTO differisce dalla TIGRE?
U: è domestico.

Da questo momento il calcolatore avrà in memoria le caratteristiche di un nuovo animale (gatto) che prima non aveva a disposizione: ha quindi acquisito esperienza nel senso che, ripetendo il gioco, rifarà le stesse domande, ma chiederà "E' un gatto?" lì dove prima aveva chiesto "Come si chiama l'animale?".

Come puoi intuire, un vero e sofisticato programma di IA che sia degno di chiamarsi "Sistema esperto" può arrivare a gradi di sofisticazione enormi con applicazioni nei campi più disparati: previsioni del tempo, dell'andamento della borsa, del risultato delle elezioni e così via.

Easy Script

□ Sono in possesso del fascicolo 41 di C.C.C. che contiene l'inserto dedicato interamente all'uso di Easy Script. Come posso trovare un maggior numero di informazioni sul potente word processor della Commodore?
(Luca Gatti - Albese)

• L'inserto cui ti riferisci era destinato a coloro che, pur possedendo il programma, non erano in grado di utilizzarlo al meglio a causa di alcuni malfunzionamenti dovuti ad imperizia d'uso.

Se, quindi, desideri entrare in possesso dell'elenco completo di tutti i comandi disponibili, non ti rimane altro che acquistare, da un qualsiasi rivenditore autorizzato, la confezione originale del software che contiene un esauriente libretto di istruzioni in italiano.

Da disco a nastro

□ Vi sono alcuni giochi su disco che sono formati da più di un programma. Ho saputo che è possibile trasferirli su cassette con alcune istruzioni Poke. E' vero?
(Maurizio Frustoni - Terranuova)

• Sì e no. Mi spiego meglio: nei programmi formati da più file, il primo di questi, in genere, contiene alcune istruzioni che provvedono ad inserire particolari codici di protezione che impediscono l'interruzione del gioco e, quindi, la possibilità di registrarlo.

All'interno dello stesso file, inoltre, sono presenti altre istruzioni che abilitano, o meno, il caricamento degli altri file presenti sullo stesso disco.

Saranno certamente presenti, da qualche parte nel programma, le istruzioni che "legano" i file tra loro o altre che sovrappongono le istruzioni necessarie ad altre che non servono più.

Siccome ogni gioco ha una sua tecnica particolare di protezione (e di fusione di file) non è possibile determinare un sistema valido universalmente, nè per realizzare un unico file nè, tantomeno, per effettuare trasferimenti su cassetta.

Modem e smembramenti

□ Ho notato che alcuni negozi di computer smembrano le recenti confezioni telematiche della Commodore e vendono separatamente i vari componenti: computer, drive, modem, Geos. Se dovessi acquistare un apparecchio proveniente da una confezione telematica, pensate che avrei problemi con la garanzia?
(Luigi Casagrande - Livorno)
(Diego Giorgi - Gela)

• La Commodore Italiana, lanciando sul mercato le due confezioni telematiche, si proponeva di offrire l'opportunità di entrare nel meraviglioso mondo della trasmissione dati e forse non immaginava che alcuni negozianti, per favorire i propri clienti, provvedessero a separare i vari componenti della confezione stessa.

Tale modo di comportarsi, a onor del vero, dipende soprattutto dal fatto che la Commodore ha deciso di commercializzare in un secondo momento il solo modem e di riservare la fornitura del Geos solo ai neopossessori del C/64.

Per la garanzia, comunque, stai tranquillo: ogni componente della confezione è dotato della sua garanzia che rimane validissima anche se hai acuistato il prodotto separatamente dagli altri.

Differenze 1

□ Che differenza c'è tra il 1541 ed il 1571? Che differenza c'è tra l'operazione di Reset e lo spegnimento (e successiva riaccensione) di un computer?
(Alessio Troyli - Roma)

• Il 1571 è un drive che, oltre ad altre caratteristiche (tra cui una maggior velocità) possiede due testine di lettura/scrittura e, di conseguenza, utilizza entrambe le facce di un disco. Naturalmente deve essere collegato ad un computer che lo utilizzi in tale modo, altrimenti funziona come un semplicissimo 1541. Il computer in questione è il C/128 che, "accorgendosi" di essere in presenza di un 1571, lo utilizza al meglio delle sue possibilità. Collegato ad un C/64, invece, si limita a simulare le funzioni del 1541.

Per ciò che riguarda le operazioni di Reset, invece, c'è da dire che il vero e totale Reset è costituito unicamente dallo spegnimento e riaccensione della macchina. In molti casi, comunque, la pressione del tasto Reset, oppure il ricorso ad una Sys particolare, provoca lo stesso risultato. Con alcuni programmi, tuttavia, (specie se protetti oppure scritti in linguaggio macchina) non ha alcun effetto.

In definitiva, vi sono tre vie per resettare un computer: la prima, più blanda, è la Sys opportuna; se questa non funziona è necessario premere il tasto Reset; in caso negativo si spegne e si riaccende la macchina.

Differenze 2

□ Che differenza c'è tra Poke X, Peek(X)And 254 e Poke X,0 e simili? Che cosa cambia, insomma, se invece di trascrivere un risultato da calcolare immetto il risultato finale?
(Pierluigi Montanari - Angera)

• Non cambia assolutamente nulla, anzi la velocità di elaborazione aumenta, dal momento che vi sono meno calcoli da effettuare. Il motivo per cui trovi spesso una forma sintattica più complicata è comunque un altro.

Molto spesso, infatti, lavorando con istruzioni Poke, risulta necessario modificare solo alcuni dei bit relativi ad una certa locazione di memoria. Le varie operazioni And e Or servono, quindi, a fare in modo che sia interessato solo quel bit e non altri.

Facciamo un esempio: sai benissimo che con Poke 1024,1 compare, in alto a sinistra dello schermo del C/64, il carattere "A"; orbene con...
Poke 1024, Peek(1024) or 128
...non fai altro che settare il settimo bit che, nel codice di schermo, rende in Reverse il carattere visualizzato. Lo stesso effetto, ovviamente, lo potresti ottenere con:
Poke 1024,129

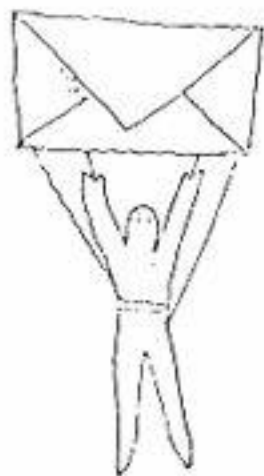
La differenza sostanziale consiste nel fatto che mentre con il valore 129 ottieni la visualizzazione in reverse del solo carattere "A", con l'operazione OR 128, invece, sei in grado di rendere in reverse qualunque carattere sia presente nella locazione video 1024.

Ovalizzazione di cerchi

□ Lavorando in alta risoluzione riesco a disegnare cerchi perfetti che però, riportati su carta con operazioni di hard copy, diventano ellissi. Quale ne è la causa?
(Aldo Manunza - Lecce)

• La cella video del C/64 è formata da una griglia di 8x8 puntini elementari

la cui distanza reciproca è la stessa sia se misurata in orizzontale sia se in verticale.

Alcune stampanti, invece, pur avendo una matrice analoga, non lasciano la stessa distanza lungo i due assi; ne consegue la deformazione lamentata che, per essere eliminata, costringe l'utente a disegnare, su schermo, ellissi che, deformate dalla stampante, ridiventano... cerchi.

Archivi universali

□ I file creati con Dbase II (C/128 in modo CP/M) possono essere utilizzati da altri database?
(Gabriele Giambonini - Svizzera)

• I file non sono altro che una sequenza di caratteri, memorizzati su disco secondo un codice particolare che è quello specifico del software adoperato.

Molti programmi, soprattutto di word processing, generano file secondo un codice praticamente universale; naturalmente alcuni comandi di formattazione, registrati spesso insieme ai file, non hanno effetto con altri W/p oppure provocano inchiodamenti del sistema.

Da un po' di tempo, però, molte software house cercano di realizzare programmi che, pur se profondamente diversi tra loro, generino file di utente che possano essere richiamati facilmente anche da altri package.

I programmi in grado di elaborare file provenienti da software professionale, comunque, riportano bene in vista, sulla confezione posta in vendita, tale opportunità.

Il problema, semmai, è un altro: bisogna essere abbastanza fortunati da trovare un package che non solo sia in grado di effettuare i caricamenti che desideri, ma, soprattutto, che utilizzi un formato software dei dischetti compatibile con quello richiesto dal drive che usi con il tuo C/128.

Stampa leggera

□ La mia Mps-803, anche con nastri nuovi, stampa caratteri flebili; è possibile far stampare due volte uno stesso rigo in modo da renderlo più leggibile?
(Ennio Balocchi - Roma)

• In linea teorica è possibile, ma sarebbe necessario un software specifico da incorporare in ogni programma.

Con i programmi professionali (W/p, data base e così via) non c'è niente da fare.

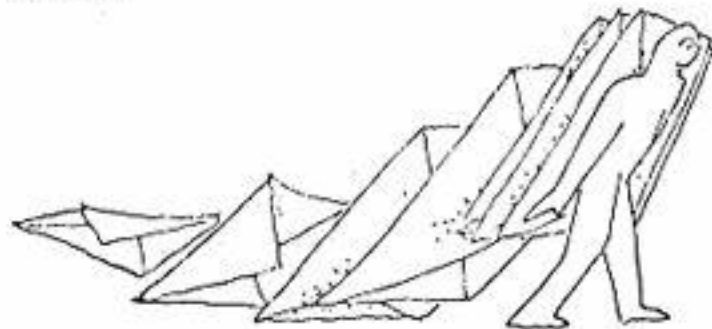
Ingrandimenti

□ Quali sono le Poke misteriose che consentono di ingrandire i caratteri del mio C/128?
(Pietro Pace - Monopoli)

• Non esistono Poke che consentano di ingrandire i caratteri visualizzabili; è infatti necessario un apposito programma che riporti su video (o su stampante) l'opportuno ingrandimento.

Le schermate, che spesso vedi in vari programmi, e che riportano messaggi di altezza e forma più o meno differenti da quelle alle quali sei abituato, sono dovute a particolari subroutine, incorporate negli stessi programmi (e difficilmente "estraibili") che servono a rendere più suggestive determinate immagini.

Solo l'Amiga, il recente computer della Commodore, permette, in alcuni casi, la selezione tra vari set di caratteri.

Lavorando con alcune stampanti sofisticate è invece possibile, con una certa facilità, ottenere set di caratteri alternativi, anche di varia grandezza.

Pet Speed C/128

□ Mi sono procurato il compilatore Pet Speed per C/128 ma, specialmente in grafica, le schermate riprodotte dal programma compilato sono molto diverse da quelle ottenute con il programma sorgente in Basic. Quale può esserne la causa?
(Federico Solla - Napoli)

• Le notizie che fornisci sono poco dettagliate: per saperne di più dovresti inviarci il dischetto contenente sia il sorgente che l'oggetto.

E' probabile che, nel programma in Basic, vi siano Poke "strane" o che, peggio, invadano l'area su cui opera il compilatore. Se questo non è il tuo caso, purtroppo, non posso esserti di alcun aiuto.

Banchi del C/128

□ Che cosa sono i banchi di memoria del C/128 (che il C/64 non possiede) e come è possibile che un microprocessore ad otto bit riesca a gestire una memoria più ampia di 64 K?
(Cristiano Martorella - V.Equense)

• Il C/128 può operare su molti banchi di memoria da 64K ciascuno. E' ovvio che molti di questi sono "vuoti" nel senso che il computer, nella sua confezione originale, ne occupa solo due; gli altri sono liberi per espansioni di Ram, varie cartucce e così via.

Non è esatto dire che il C/64 non disponga di più banchi. In effetti il C/64 ha due banchi di memoria a sua disposizione, benchè non utilizzi il secondo per intero. Si pensi, infatti, che sono presenti 64K Ram e, parallelamente, "coesistono" altri K di memoria Rom (per non parlare dei Chip I/O e simili).

Per selezionare uno dei vari chip, che sono costretti a condividere gli stessi indirizzi (ma che devono funzionare selettivamente, uno solo alla volta), il microprocessore invia un particolare segnale che abilita uno solo dei chip mentre, contempora-

128 KBYTES

SINCLAIR COM

COMMODORE

MSX

DI RIVISTA.

PUTER

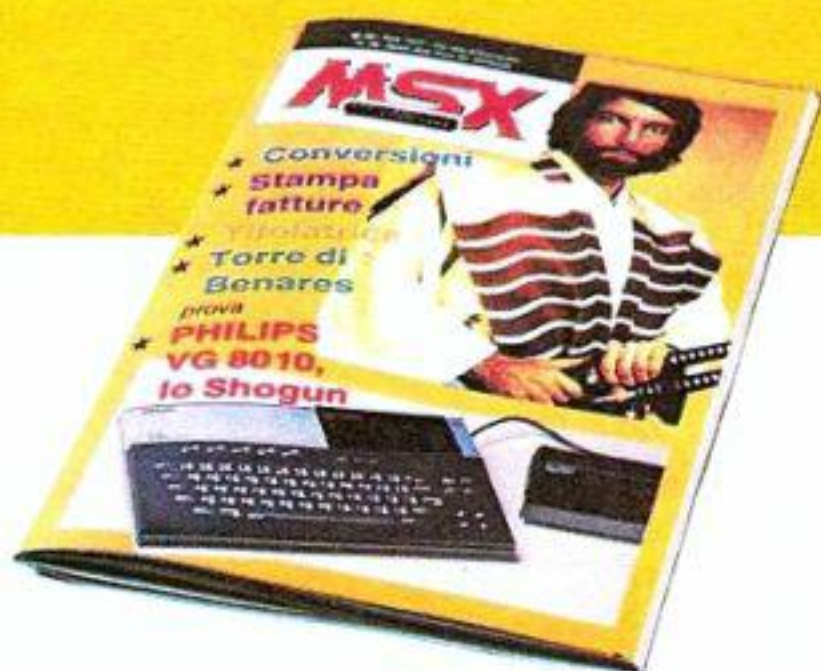
Personal computer

STUDIO DI FUNZIONE

RILOCATORE DI PROGRAMMI

FUNZIONE VAL PER IL QL

TRE RIVISTE IN UNA!

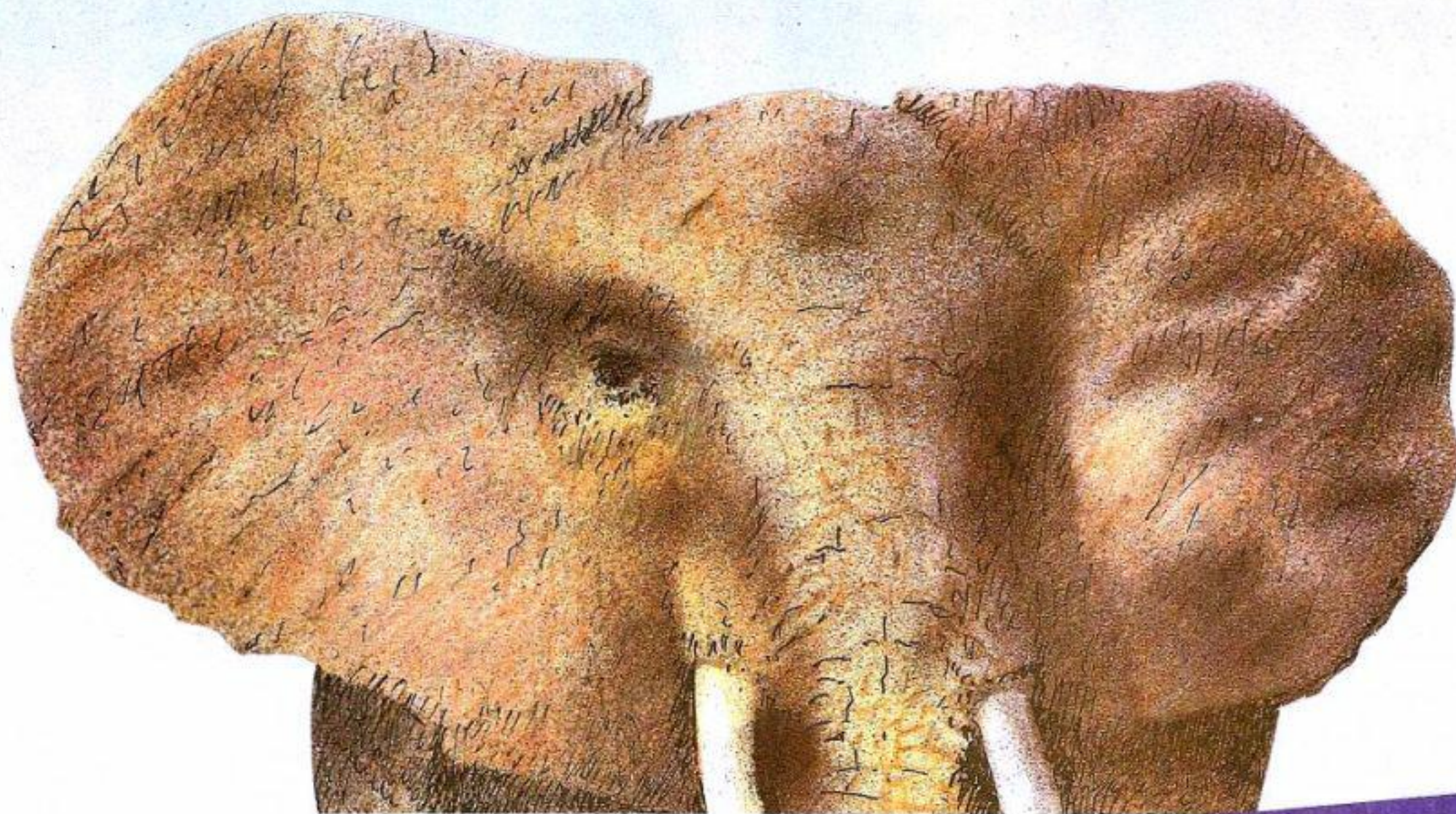
E' IN EDICOLA

Personal computer è la rivista Systems per gli utenti Commodore, MSX, Sinclair. Non solo tre riviste per tre diversi utenti: **Personal Computer** è anche un'idea nuova per far comunicare tutti gli hobbisti. **Personal Computer:** 128 Kbytes di rivista, tutti i mesi in edicola. L'abbonamento cumulativo a **Computer** e **Personal Computer** costa solo L. 65.000.

Il mercato si evolve. Anche noi.

IMPORTATRICE DISTRIBUTRICE
IN ESCLUSIVA PER L'ITALIA

ETP SRL
ELECTRONIC AND TECHNICAL PRODUCTS
via del macao, 4 - 00185 roma
tel. 4743080 - 4746880

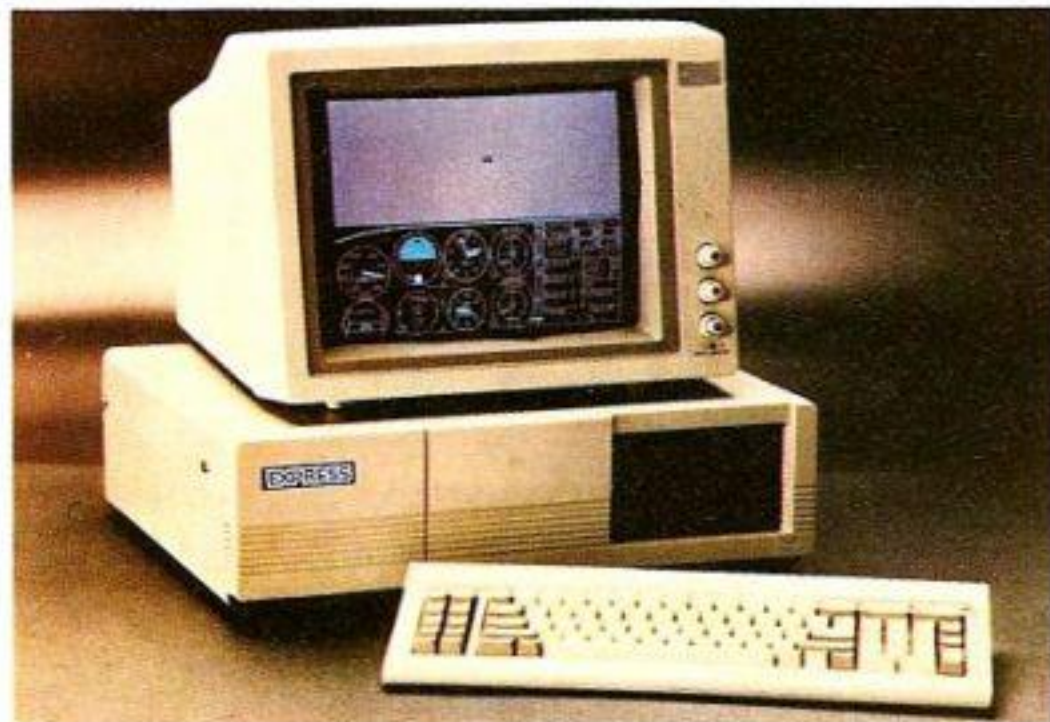
**RITIRIAMO IL VECCHIO
CONSEGNAMO IL NUOVO**

Una volta... nel branco...

Oggi

XT
- Totalmente compatibile PC/XT IBM
- microprocessore Intel 8088
- memoria RAM 256 Kb espandibile a 640 Kb sulla piastra base
- memoria ROM 8 Kb (BIOS) espandibile 64 Kb
- scheda madre con 8 slot di espansione
- scheda grafica monocromatica ad alta risoluzione (720x348 punti)
- porta parallela per collegamento stampante
- tastiera italiana ASCII con 84 tasti e 10 tasti funzione o USACII
- uno o due drive slim 5"1/4 da 360 Kb (o disco rigido di diverse capacità)
- alimentatore 135 W - 220 Volt
- dimensioni: 500 x 410 x 142 mm
- peso: 11 Kg
- sistemi operativi supportati: tutti quelli del PC/XT IBM

AT
- Totalmente compatibile PC/AT IBM
- microprocessore Intel 80286: trasferimento dati a 16 bit; indirizzamento a 24 bit
- memoria RAM 512 Kb espandibile a 3 MB
- scheda madre con 8 slot di espansione
- scheda grafica monocromatica ad alta risoluzione (720x348 punti)
- porta parallela per collegamento stampante
- spie luminose sulla tastiera per "capss lock", "num lock" e "scroll lock"
- orologio, calendario e configurazione di sistema con CMOS RAM e relativa alimentazione di riserva con batteria interna
- serrattura di sicurezza per l'accesso all'unità di elaborazione
- drive Floppy disk 1.2 Mb

Massima espandibilità con dischi e schede

studio A R I 06/490568

neamente, "addormenta" tutti gli altri.

La gestione di tale procedura, oltre ad essere piuttosto complessa, risulta, in ultima analisi, molto lenta ed è questo il motivo per cui, nei computer moderni, si preferisce ricorrere ad altri microprocessori in grado di indirizzare i vari dati senza problemi di selezione di banchi.

Titolatrice

☐ Perchè non provvedete a scrivere un programma che visualizzi scritte scorrevoli, multicolori e di varia grandezza che possa essere adoperato come titolatrice per i videoregistratori?
(Antonio Polimeno - Matino)

• Molto tempo fa apparve un mio programma sulla rivista Micro & Personal Computer (N.39 marzo '84) che consentiva di riprodurre su video messaggi piuttosto ingranditi.

Per quell'epoca, forse, il programma era sufficientemente valido ma, con il passare del tempo, sono venuti alla luce numerosi altri software che consentono di visualizzare messaggi di notevole eleganza.

Il ricorso ai vari Print Shop, Print Master e simili può rappresentare, pur se non sono stati creati per lo scopo indicato dal nostro lettore, una valida opportunità per abbellire e personalizzare le videoregistrazioni.

Se qualcuno ritiene di aver scritto un programma sufficientemente interessante, può inviarcelo per il suo inserimento in "Directory", il dischetto che vendiamo mensilmente solo per corrispondenza.

Rilocare programmi in L.M.

☐ Come si può rilocare un programma scritto in linguaggio macchina?
(Piergiorgio Dell'Agostin - Belluno)

• La teoria della tecnica di rilocazione sembrerebbe piuttosto semplice: individuare le istruzioni relative ad indirizzi assoluti e modificare l'argomento in modo che "punti" al nuovo indirizzo.

Passando dalla teoria alla pratica, però, la faccenda si complica paurosamente, ed indico qui di seguito solo alcuni dei motivi:
- necessità di distinguere tra indirizzi assoluti da mantenere inalterati (come quelli relativi alle Rom) e quelli che, invece, appartengono alla routine l.m. (e, quindi, da modificare).
- necessità di individuare, all'interno del programma, la presenza di eventuali tabelle, da lasciare, ovviamente, inalterate.
- necessità di considerare eventuali sovrapposizioni nei casi in cui si voglia rilocare di pochi indirizzi un programma di lunghezza maggiore del "salto" impostato.

In definitiva è molto, molto complicato rilocare un programma l.m. di cui non si possieda il disassemblato commentato.

La faccenda cambia totalmente se, invece, il programma l.m. è scritto dall'utente stesso. Sono oggi disponibili programmi di notevole sofisticazione (Merlin, Macroassembler e tanti altri) che consentono di scrivere un programma in Assembly e presentano facilitazioni di tutto rispetto tra cui, appunto, la rilocazione, l'inserimento di nuove istruzioni, l'allocazione di tabelle, il calcolo automatico di salti condizionati e non, l'uso delle Label.

Oggi, insomma, nessuno lavora più in linguaggio macchina "puro" e l'uso di uno strumento più sofisticato (come, appunto, gli assemblatori) è praticamente indispensabile per chi intenda lavorare in l.m. senza difficoltà eccessive.

Strano errore

☐ Con il seguente programma...:
10 For H=1 to 16
20 Poke H,40
30 Next H
...il mio C/16 indica "Syntax error in 30". Che cosa succede esattamente quando nella locazione 16 viene immesso il valore 40?
(Luca Acelli - Villanovezza)

• Le prime locazioni di memoria dei computer Commodore sono riservate al Sistema Operativo ed all'interprete Basic; l'utente non dovrebbe mai modificarne il contenuto.

Spiegare esattamente che cosa succede è un po' problematico; mi limiterò a dire che le locazioni citate servono al tuo C/16 come riferimento per le varie elaborazioni. E' intuitivo che se provvedi a "forzare" il loro contenuto, il computer non si raccapezza più e si comporta in modo strano, emettendo anche segnalazioni di errori che, in realtà, non sono presenti nel listato.

Musicalmente

☐ Tento da molto tempo di scrivere un programma che consenta di eseguire musica a più voci con il mio C/64, ma trovo molta difficoltà a completarlo.
(Alessandro Scrigner, dodicenne di Trieste)

• E ti credo! Un programma del genere dovrebbe essere scritto in linguaggio macchina da chi conosce alla perfezione non solo le peculiarità delle note musicali, ma anche la gestione del suono con il C/64.

Ti consiglio di utilizzare uno dei tantissimi Music Editor che circolano tra gli appassionati, oppure di studiare con molta attenzione tutti gli articoli che finora sono apparsi sulla nostra rivista.

Miracoli

☐ Vorrei sapere se con il vostro emulatore Gw-Basic è possibile emulare anche le funzioni del computer IBM 370.
(nome omesso per carità cristiana)

• Avevamo intenzione di realizzare qualcosa del genere, ma abbiamo incontrato difficoltà nel richiedere consigli specifici al Padreterno.

Risposte Rapide

Problemi con Header

Ho provato ad adoperare la sintassi che tu hai utilizzato nella formattazione dei dischetti, ma non ho incontarto inconvenienti di sorta. Sei sicuro che, sullo stesso rigo del comando, non fossero presenti altri caratteri che generino il Syntax Error lamentato?

(Antonio Merchiori - Mestre)

Archivio codificato

Ho già avuto modo di ricordare, in una precedente risposta, che per usare correttamente il programma GE. DA.FI. (apparso su C.C.C. N.37) è indispensabile che almeno un campo sia codificato.

Ne approfitto per suggerire, a tutti i lettori, di leggere SEMPRE questa rubrica: molto spesso troverete risposte ai vostri dubbi.

(Ivo Neri - Pontedera)

Software professionale

Per procurarsi il software professionale (e altro software in generale), ti consiglio di rivolgerti ai nostri inserzionisti che vendono software per corrispondenza, oppure a punti di vendita di tua fiducia.

(Paolo D'Agapiti - Velletri)

Non per C/16

I programmi dell'articolo "Sprite istruttivi" apparso sul N.41, possono girare solo sul C/128 perchè si basano sulla gestione degli sprite che il C/16 (come il Plus/4) non possiede.

(Massimo Bregant - Gorizia)

Leggere, leggere, leggere

Stavolta sono cattivo: il metodo per determinare via software se una periferica è accesa oppure spenta è già stato descritto in questa stessa rubrica nei numeri 40 e 42; ed è anche usato nel programma "Botte piena..." pubblicato sul N.40. Morale: rileggi gli arretrati (che assicuri di possedere) e, visto che ci sei, leggi anche gli altri articoli che hai trascurato.

(Norberto Baroni - Vercurago)

Software offresi

A te, come a tutti gli altri lettori in gamba, non posso che rinnovare l'invito a telefonare (02/84.67.34.8) per stabilire se il software prodotto risponde alle nostre esigenze.

(Andrea - Somewere)

H/copy C/128

Sul N.43 della nostra rivista è apparso un programma (Quark) che contiene una routine in L.M. per la riproduzione su carta delle schermate in alta risoluzione del C/128 in modo 40 colonne.

Per ottenere l'hard copy nel modo 80 colonne puoi utilizzare il nostro software "Graphic expander" venduto su disco solo per corrispondenza che, oltre alla opzione citata, raggruppa numerosi comandi Basic per la gestione della grafica in alta risoluzione del C/128 in modo 80 colonne.

(Piero Emanuel - Giovinazzo)

Il nostro software

Il software che noi offriamo ai nostri lettori è solo quello descritto nelle pagine pubblicitarie che compaiono su C.C.C. Per procurarti i programmi che non fanno parte del nostro elenco devi rivolgerti ai negozi specializzati.

(Alessandro Tingali - Lentini)

Programma carino

Il programma che consente di riportare su carta l'output che hai inviato è simpatico e interessante. Contattaci telefonicamente per fissare un incontro.

(Matteo Castelli - Monza)

Modem Commodore

Il modem Commodore diffuso recentemente nei punti di vendita autorizzati può essere applicato solo al C/64 e al C/128. Non può quindi funzionare sugli altri computer.

(Pasquale Scalercio - Roma)

Hai fatto bene

Non preoccuparti: il tuo ultimo acquisto (C/128) è un ottimo computer, pur se un po' trascurato dalle software house. Con il C/128 è possibile utilizzare tutto il software disponibile per il C/64 e, ovviamenete, potrai collegare qualsiasi accessorio valido anche per il C/64.

P.S: Grazie per la cartolina della tua bella città!

(Gianluca - Vieste)

Speed Dos e 1541/C

Prima di acquistare il set di speed dos, ti consiglio di descrivere accuratamente gli apparecchi in tuo possesso al negozio cui ti vorrai rivolgere.

(Matteo Grande - Sottomarina)

C/16 difettoso

Penso che al tuo C/16, che si comporta sempre in modo strano dopo una mezz'ora di funzionamento, farebbe bene una visitina presso un centro di riparazioni specializzato.

(Fabio Quinci - Roma)

Illegal quantity

Sono pronto a scommettere che hai sbagliato a digitare alcuni dei valori numerici, presenti nelle righe Data del programma T/Tape C/128 (C.C.C. n. 42).

La segnalazione "Illegal quantity error", infatti, viene visualizzata quando si tenta di Pokare un valore maggiore di 255 oppure negativo.

(Alessandro Agatea - Saccolongo)

Strani comportamenti

Lo strano modo di operare del programma T/Tape C/128 (C.C.C. n.42) dipende certamente dalla errata trascrizione dei valori numerici presenti nelle righe Data.

(Sandro Giangreco - Enna)

Già pubblicato

Il breve programma che hai inviato svolge le stesse funzioni di un altro listato che abbiamo già pubblicato in precedenza.

(Andrea Preziosi - Somewere)

Quante vite?

Non esiste una locazione "speciale" cui riferirsi per determinare il numero di vite a disposizione durante lo svolgimento dei videogame.

(Giandonato Pappalardo - Barletta)

Club
Tra breve pubblicheremo tutte le notizie relative a quei club di informatica che ci invieranno su dischetto le informazioni richieste .
(Alberto Bianchi - Cuneo)

Vic 20
Non dedichiamo spazio al Vic 20, non perchè è uscito di produzione, ma perchè non vi sono più lettori che chiedono di parlarne.
(Carlo Breggi - Siena)

Rozzano Club
Non posso pubblicare l'indirizzo del tuo Club per il semplice motivo che hai dimenticato di inviarlo(!)
(Gennaro Gaetano - Rozzano)

Buon programma, ma...
I programmi che hai inviato su carta sono (forse) interessanti, ma, per la loro pubblicazione, è indispensabile che pervengano su disco, corredati da relativo articolo esplicativo.
(Marco Baraldi - Adria)

Ma come hai fatto?
Non riesco proprio a capire come tu possa aver caricato su C/16, dotato di solo registratore, il lungo programma Ge.Da.Fi. apparso sul N.37 che richiede esclusivamente il drive per poter funzionare correttamente.
Forse è questo il motivo dei malfunzionamenti di cui ti lamenti.
(Mauro Montana - Pisa)

Indici ragionati
Al più presto pubblicheremo sistematicamente l'elenco degli articoli e dei programmi apparsi sulla nostra rivista.
(Mauro Strinna - Cagliari)

Troppe domande
Mi dispiace, ma il numero di domande che poni nella (simpaticissima) lettera mi hanno scoraggiato al punto di non sapere che cosa rispondere; perdonami!
(Angelo D'Amico - Cava dei Tirreni)

Programmi validi
I programmi che hai inviato su disco mi sembrano interessanti ma sono privi dell'articolo esplicativo, indispensabile per proporli ai nostri lettori: telefonami.
(Teo Usuelli - Roma)

Fuori l'autore
L'autore degli inserti dedicati al CP/M disponibile sul C/128 (apparsi sulla rivista "Personal Computer") è Pietro Rozzo. Mi scuso per l'errore di trascrizione dovuto ad una serie di involontari equivoci.

Pippo
Il file sequenziale dal nome "Pippo", lungo 66 blocchi (e presente in un dischetto di Directory), è il file elaborato dal programma "Calc Result" e rappresenta la tabella riportata nell'inserto "Come usare correttamente Calc Result" contenuto nel N.42 della nostra rivista.
(da alcune telefonate)

Directory

Tutti i programmi pubblicati su questo numero di Commodore Computer Club, sono registrati su un dischetto appartenente alla serie "Directory".

Oltre ai programmi citati, sono presenti altri file di notevole interesse per coloro che desiderano realmente utilizzare il proprio computer.

Sono infatti presenti, di solito, anche i programmi pubblicati sull'altra nostra rivista "Personal Computer", ed altri file che, in totale, riempono quasi per intero i 664 blocchi normalmente disponibili su un floppy disk.

Sono disponibili i seguenti dischetti:

***Directory N. 1** (CCC N.34 + raccolta dell'intera Enciclopedia di routine)*
***Directory N. 2** (CCC N.35 + CCC N.36)*
***Directory N. 3** (CCC N.37 + PC N.7 + PC N.8)*
***Directory N. 4** (CCC N.38 + file vari)*
***Directory N. 5** (CCC N.39 + PC N.9)*
***Directory N. 6** (CCC N.40 + PC N.10)*
***Directory N. 7** (CCC N.41 + file vari)*
***Directory N. 8** (CCC N.42 + file vari)*
***Directory N. 9** (CCC N.43 + file vari)*
***Directory N. 10** (CCC N.44 + file vari)*

Come procurarsi i dischetti della serie "Directory"

Avvertiamo i lettori che NON è assolutamente possibile inviare i programmi su nastro, per intuibili motivi di economia ed affidabilità del nastro cassetta.

Ogni numero di "Directory" può quindi esser richiesto SOLO su disco inviando L.12000 per ciascun disco oltre a L.3000 (fisse) per le spese di imballo e spedizione (indipendenti dal numero di dischi richiesti).

Chi desiderasse la spedizione raccomandata, deve aggiungere altre 3000 lire per l'ulteriore affrancatura.

Non ci è possibile inviare materiale contrassegno: si prega di astenersi dal chiedere eccezioni alla regola.

Compilate un normale modulo di C/C postale indirizzando a:

C/C postale N. 37952207
Systems Editoriale
Viale Famagosta, 75
20142 Milano

Non dimenticate di indicare chiaramente, sul retro del modulo (nello spazio indicato con "Causale del versamento") non solo il vostro nominativo completo, ma anche il numero del disco desiderato; esempio:

"Directory N.1"
"Directory N.3"
"Directory N.4"

Totale:
L.12000x3 +
L.6000 (spediz. racc.)
= L.42000

(spese di imballo e spediz. racc. comprese).

N.B. Per ottenere il materiale ordinato in tempi più ristretti, inviate l'importo a mezzo assegno bancario non trasferibile con lettera di accompagnamento: le poste italiane non brillano per velocità! (due mesi circa per il recapito di un C/C postale).

Un monitor affidabile

Non sempre un monitor a colori rappresenta il meglio per i package professionali; per non parlar dell'aspetto economico...

di Alessandro de Simone

Molte lettere che riceviamo tornano spesso sull'eterno dilemma: è meglio procurarsi un monitor a colori, un TV oppure un semplice bianco e nero?

La risposta è tutt'altro che semplice perchè, per rispondere correttamente, è indispensabile sapere l'uso che si deve fare del sistema di visualizzazione.

Tuttavia possiamo riportare una scala di valori che rappresenta, a seconda dei casi, lo schermo ideale:

• TV color domestico. Questo è sicuramente l'optimum dal punto di vista dell'economia: è statisticamente dimostrato che, nelle case in cui entra un computer, è presente anche un apparecchio TV color; quasi sempre, però, questo ha dimensioni piuttosto grandi e lo rende idoneo solo per videogiochi e software di intrattenimento. Assolutamente sconsigliato per lavori di word processing e professionali in genere, ove è richiesta una grande nitideza ed una dimensione piccola del video.

• Monitor colore specifico per computer. La qualità, rispetto alla soluzione precedente, aumenta di parecchi punti sia per le ridotte dimensioni dello schermo, sia per la maggiore nitidezza disponibile. Solo la riproduzione sonora lascia, spesso, a desiderare.

• Monitor B/N specifico per computer. E' l'deale per lunghi lavori con software professionale. Non è possibile realizzare, infatti, un video a colori di nitidezza superiore a quello in bianco e nero; ne consegue una visualizzazione di tutto riposo che non affatica la vista anche dopo molte ore di lavoro. Non per niente, negli uffici, trovate quasi esclusivamente monitor monocromatici.

I modelli Philips

La Phiiips, industria leader nel campo dell'elettronica, offre sul mercato anche modelli in b/n, di alta affidabilità e, soprattutto, basso costo.

I monitor ai quali ci riferiamo sono dotati di schermo antiriflesso, di staffa di inclinazione e di cavetto con spina DIN a sei poli (ma sono disponibili, a parte, tutti i cavetti di adattamento possibili ed immaginabili). In alcuni modelli è presente la presa per la cuffia, mentre in tutti gli altri possiamo trovare i soliti comandi: volume, luminosità, contrasto, regolazione rapporto orizzontale / verticale. Un utile tasto consente di selezionare due tonalità cromatiche per lo sfondo.

La serie di modelli prevede, come optional, un piede di supporto per orientare ed inclinare il monitor nelle varie direzioni, un parasole per i sistemi che dovessero venire installati nei pressi di una finestra, un portacarta, da posizionare ai lati del monitor stesso, per alloggiare fogli dattiloscritti e facilitarne l'eventuale trascrizione.

Un larghezza di banda di 22 Mhz, e la possibilità di visualizzare fino a 2000 caratteri (schermi di 80x25) non lasciano dubbi sulla nitidezza delle immagini che i monitor Philips sono in grado di riprodurre. Il consumo, come intuitivo, è irrisorio: solo 32 watt.

Una cedola importante

Nelle nuove confezioni dei prodotti Commodore è presente uno stampato che consente di abbonarsi a condizioni vantaggiose

Chi ha appena acquistato un prodotto originale Commodore negli ultimi tempi, avrà sicuramente trovato un depliant, simile a quello riprodotto in queste pagine, che precisa alcune utili cose.

Anzitutto permette, al neo-acquirente, di sottoscrivere un abbonamento alla nostra meravigliosa (scusate l'immodestia) rivista a condizioni particolarmente vantaggiose.

Inoltre, come già era del resto avvenuto in precedenza, ribadisce che Commodore Computer Club è la rivista ufficiale della Commodore Italiana e, come tale, è il "canale" preferenziale per la diffusione di notizie relative al mondo Commodore.

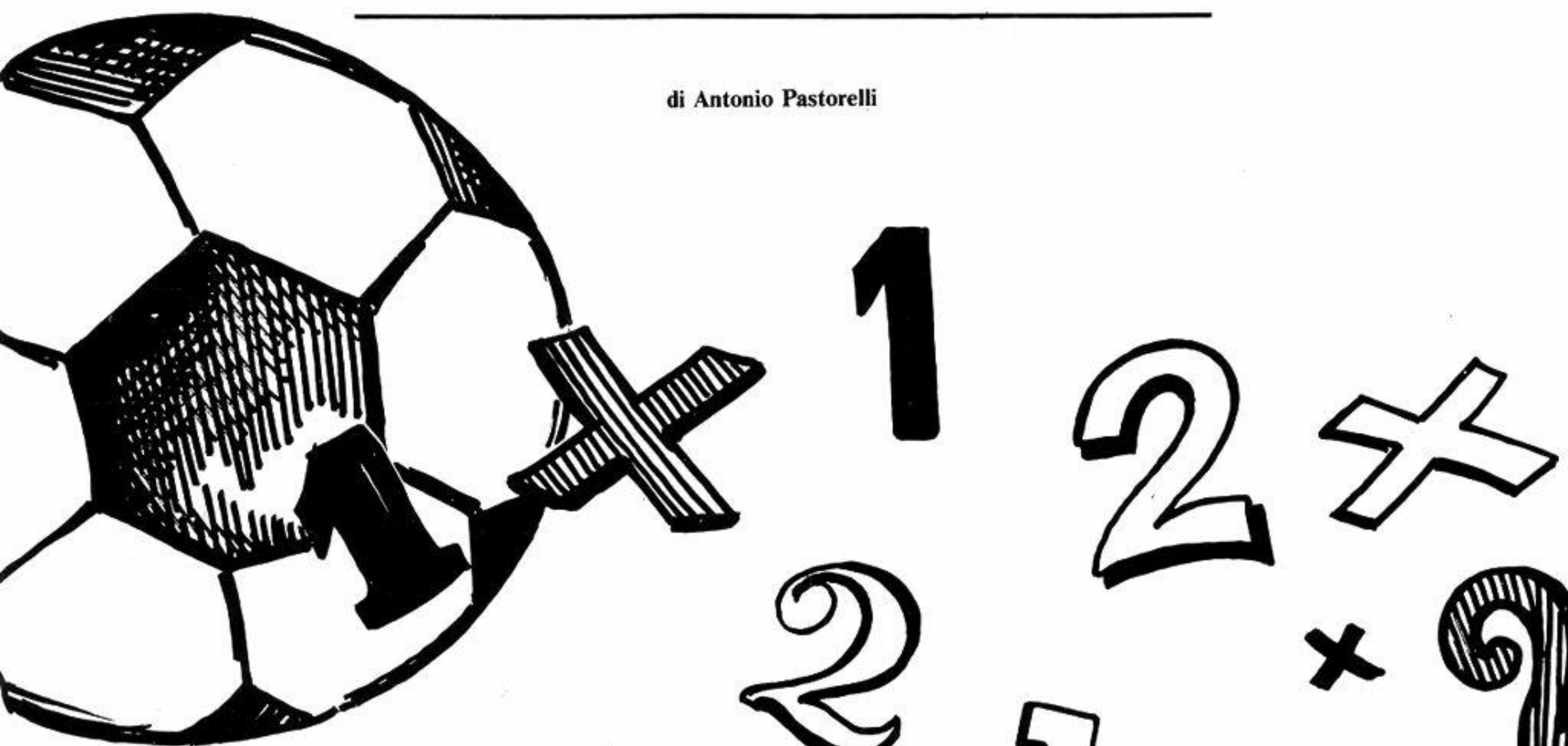
Non si può negare, infatti, che da molto tempo (direi da sempre) la nostra pubblicazione è sempre stata dalla parte dell'utente con varie iniziative: diffusione di notizie, divulgazione di programmi, articoli e recensioni che riguardassero da vicino il mondo dei computer Commodore.

E se il drive, la stampante, il computer o il software acquistato non possiede la scheda di cui parliamo?

Bè, se vi interessa l'abbonamento a condizioni privilegiate, inviate una fotocopia della scheda di garanzia allegata alla confezione: faremo uno strappo alla regola!

Pronostici Totocalcio

Come affidare al computer l'ingrato compito di valutare le probabilità di vittoria delle squadre di calcio.

di Antonio Pastorelli

Finora abbiamo presentato programmi per lo sviluppo di sistemi condizionati, ridotti, biridotti ed altri, lasciando al lettore il compito di stabilire il sistema-base da impostare per le opportune elaborazioni.

Ciò significa che il sistema-base è affetto, inevitabilmente, da influenze soggettive.

In questo articolo proponiamo un metodo per stilare un sistema, esente da tali influenze, basato soltanto su dati di fatto.

Il lettore potrà, poi, confrontare il sistema base, elaborato dal programma di queste pagine, con quello che avrebbe impostato senza "suggerimenti" di sorta: in ogni caso valuterà la schedina dopo un'opportuna riflessione e, magari, eventuali modifiche.

I dati ai quali si riferisce il programma, sono relativi ai risultati delle partite precedenti, divisi tra risultati ottenuti in casa ed in trasferta.

In base a tali informazioni, il computer calcola la probabilità che si verifichi il segno 1, X, oppure 2 e, di conseguenza, propone un sistema di fisse, doppie e triple.

Come calcolare le probabilità

Riferendoci ad un esempio pratico, supponiamo di voler calcolare le probabilità per l'incontro tra due squadre, dal nome poco fantasioso di

"Prima" e "Seconda".

Leggendo un quotidiano del lunedì, troviamo i seguenti (ipotetici) dati:

Squadre In casa Trasferta
V N P V N P

Prima 6 2 0 — — —
Seconda — — — 0 4 4

Dapprima si trasformano tali dati in percentuali:

Prima

Vinte 75%
Pareggi 25%
Perdite 0%

Seconda

Vinte 0%
Pareggi 50%
Perdite 50%

Un'alta percentuale di vincite in casa, per la squadra ospitante, abbinata ad un'alta percentuale di sconfitte in trasferta per la squadra ospite, rende "quasi" certa la vittoria della squadra che gioca in casa (segno "1"); analogo è il ragionamento nel caso opposto.

Pertanto la percentuale di vincite in casa, della prima squadra, viene moltiplicata per la percentuale di sconfitte in trasferta della seconda squadra:

Segno "1":
*75*50 = 3750*

Le percentuali di risultato nullo (pareggio), si moltiplicano tra loro:

Segno "X":
*50*25 = 1250*

Infine, si moltiplica la percentuale di perdite della prima squadra, con quella di vincite della seconda:

Segno "2":
*0*0 = 0*

L'ultima fase consiste nel riportare, su base percentuale, i risultati ottenuti...

100/(prob."1"+ prob."X"+ prob."2") = coeff.

...e, successivamente, moltiplicare "coeff.", per le singole probabilità.

Nel nostro esempio avremo, pertanto:

100/(3750+ 1250+ 0)= 0.02

*"1": 0.02*3750 = 75*
*"X": 0.02*1250 = 25*
*"2": 0.02*0 = 0*

Totale = 100

Tenendo presenti tali valori, si decide se giocare una fissa, una doppia oppure una tripla.

Il programma

Dopo aver dato il RUN, premete un tasto qualsiasi e, alla domanda "Inserisci l'elenco delle squadre?", rispondete con "S", se intendete inserirlo, "N" in caso contrario.

Se decidete di inserire i nomi delle squadre, il calcolatore chiederà, partita per partita, i nomi delle squadre contendenti.

Se non inserite l'elenco delle squadre, verranno subito chiesti i risultati, in casa per la squadra ospitante, ed in trasferta per l'altra.

tali dati sono reperibili su qualsiasi quotidiano del lunedì; tuttavia è consigliabile utilizzare i periodici specializzati perchè indicano i risultati, distinti in casa e trasferta, anche per la serie B e C.

E' facile constatare che, per una qualsiasi squadra, i risultati in casa differiscono notevolmente dai risultati ottenuti in tasferta; utilizzando, quindi, dati globali, che non tengano conto di questa distinzione, si otterrebbero previsioni non attendibili.

In particolare, il programma pone le seguenti domande, riferite alla squadra ospitante:

- *vincite in casa?*
- *pareggi in casa?*
- *perdite in casa?*

e, per l'altra squadra, analoghe domande, riferite ai risultati ottenuti in trasferta.

Terminata la noiosa fase della digitazione, sullo schermo appare l'elenco delle squadre (se inserito precedentemente), seguito dalle percentuali di probabilità per i tre possibili risultati e, infine, il sistema elaborato in base alle percentuali citate.

Nel caso vi accorgiate di aver commesso errori di battitura durante la fase di Input, basterà premere il tasto F3 e indicare il numero del pronostico i cui dati devono essere corretti.

A questo punto appaiono sullo schermo le informazioni relative al pronostico, ed il seguente menù:

1) cambia nomi squadre
2) cambia dati
3) nulla

Premendo il tasto "1", si ha la possibilità di cambiare, o modificare, il nome delle squadre (nel caso sia stato inserito), mentre premendo "2", è possibile modificare i dati relativi ad una o entrambe le squadre.

Infine, con il tasto "3", si ottiene di nuovo la visualizzazione dell'elenco delle squadre, dei pronostici e del sistema (che risulterà diverso dal precedente, nel caso abbiate apportato modifiche ai dati di uno o più pronostici).

Un listato di utilità

Alcuni lettori hanno chiesto di poter riversare su stampante le colonne del file generato con il programma TOT 13 (apparso su C.C.C. n.35).

Il secondo listato pubblicato in queste pagine, risolve il problema esposto.

SCHEDA TECNICA

Software applicativo per:
giochi d'azzardo

Idoneo per computer: C/64

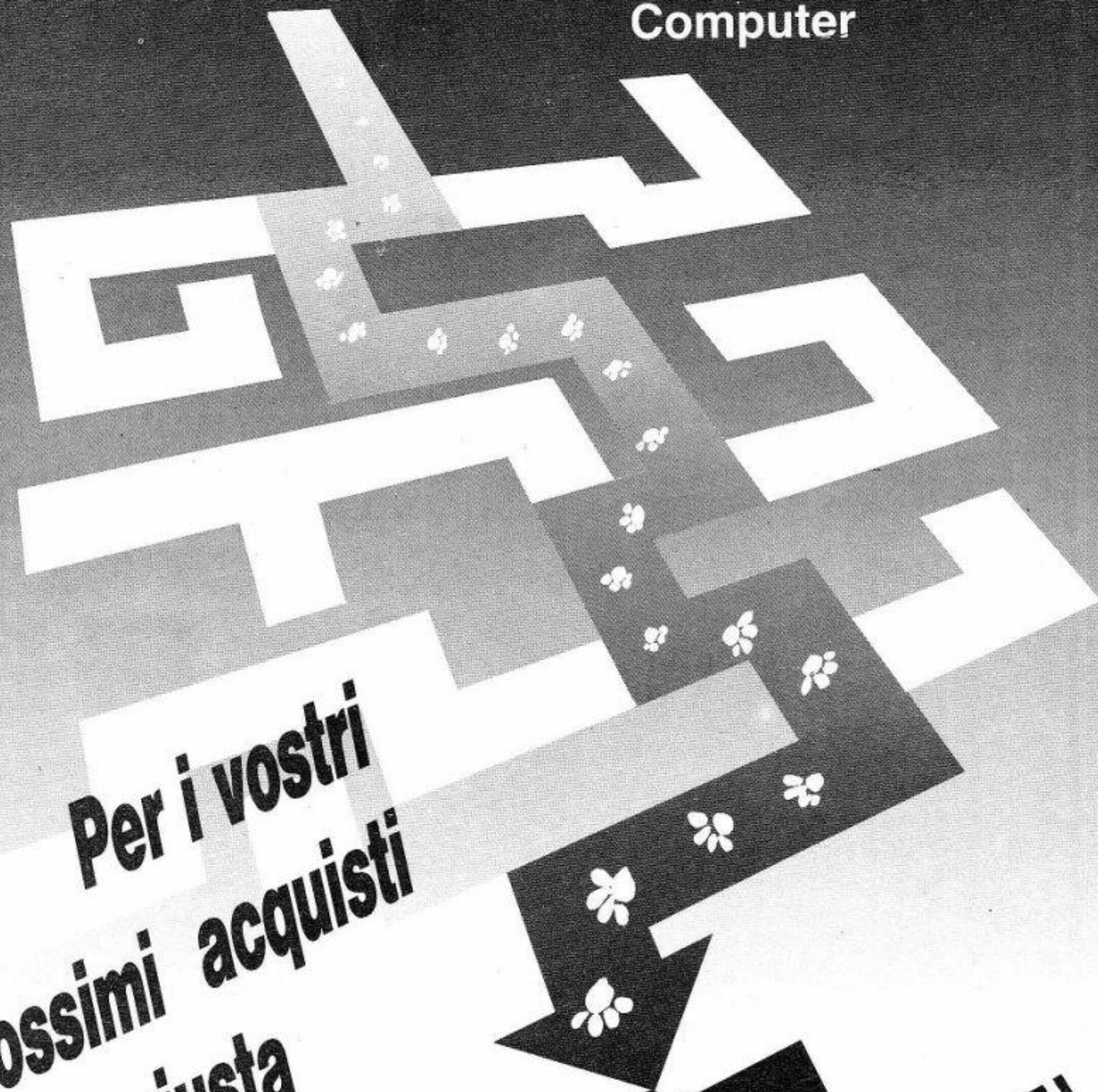
Facilmente adattabile ad altri computer Commodore

Consigliato a tutti i lettori

```
10 REM PRONOSTICI TOTOCALCIO
20 :
30 REM PER C-64
40 :
50 REM BY ANTONIO PASTORELLI
60 :
70 Z1$="SQUADRA OSPITANTE":Z2$
   ="SQUADRA OSPITE"
80 X1$=CHR$(213):X2$=CHR$(192)
   :X3$=CHR$(178):X4$=CHR$(201
   ):X5$=CHR$(219)
90 X6$=CHR$(221):X7$=CHR$(171)
   :X8$=CHR$(179):X9$=CHR$(32)
100 XA$=CHR$(202):XB$=CHR$(177)
   :XC$=CHR$(203):FOR J=1 TO 7
   :XD$=XD$+X2$:NEXT
110 Z3$="VINCITE":Z4$="PAREGGI"
   :Z5$="PERDITE":Z6$=" IN CAS
   A":Z7$=" FUORI CASA"
120 DIM A1$(13),A2$(13),V1(13),
   V2(13),X1(13),X2(13),P1(13)
   ,P2(13)
130 DIM S1(13),SX(13),S2(13):CL
   $=CHR$(147):RV$=CHR$(18):RO
   $=CHR$(146)
140 SU$=CHR$(145):GI$=CHR$(17):
   HO$=CHR$(19):HG$=HO$+GI$+GI
   $+GI$
150 PRINTCL$RV$X9$X9$X9$X9$"CAL
   COLO PROBABILITA' TOTOCALCI
   O"X9$X9$X9$X9$
160 NR=13
170 GET A$:IF A$="" THEN 170
180 PRINTCL$"INSERISCI L'ELENCO
    DELLE SQUADRE?"
190 GET A$:IF A$="S" THEN SQ=1:
   GOTO 220
200 IF A$="N" THEN 270
210 GOTO 190
220 FOR J=1 TO NR
230 PRINTCL$"PRONOSTICO";J;GI$
240 PRINTZ1$;:INPUT A1$(J):IF A
   1$(J)="" THEN PRINTSU$;:GOT
   O 240
250 PRINTZ2$;:INPUT A2$(J):IF A
   2$(J)="" THEN PRINTSU$;:GOT
   O 250
260 NEXT:PRINTCL$:IF SQ=1 THEN
   400
270 FOR J=1 TO NR
280 PRINTCL$RV$"PRONOSTICO";J;G
   I$
290 PRINTZ3$Z6$" "Z1$;:INPUT V1
   (J):IF V1(J)<0 THEN PRINTSU
   $;:GOTO 290
300 PRINTZ4$Z6$" "Z1$;:INPUT X1
   (J):IF X1(J)<0 THEN PRINTSU
   $;:GOTO 300
310 PRINTZ5$Z6$" "Z1$;:INPUT P1
   (J):IF P1(J)<0 THEN PRINTSU
   $;:GOTO 310
320 IF V1(J)=0 AND X1(J)=0 AND
   P1(J)=0 THEN 280
330 PRINTGI$GI$
340 PRINTZ3$Z7$" "Z2$;:INPUT V2
   (J):IF V2(J)<0 THEN PRINTSU
   $;:GOTO 340
350 PRINTZ4$Z7$" "Z2$;:INPUT X2
   (J):IF X2(J)<0 THEN PRINTSU
   $;:GOTO 350
360 PRINTZ5$Z7$" "Z2$;:INPUT P2
   (J):IF P2(J)<0 THEN PRINTSU
   $;:GOTO 360
370 IF V2(J)=0 AND X2(J)=0 AND
   P2(J)=0 THEN PRINTSU$SU$SU$
   ;:GOTO 340
380 GOSUB 1780:IF RP=1 THEN 280
390 NEXT:GOTO 640
400 FOR J=1 TO NR
410 PRINTCL$RV$"PRONOSTICO";J;G
   I$
420 PRINTRV$A1$(J)":"RO$Z3$Z6$;
   :INPUT V1(J)
430 IF V1(J)<0 THEN PRINTSU$;:G
   OTO 420
```

```
440 PRINTRV$A1$(J)":"RO$Z4$Z6$;
    :INPUT X1(J)
450 IF X1(J)<0 THEN PRINTSU$;:G
    OTO 440
460 PRINTRV$A1$(J)":"RO$Z5$Z6$;
    :INPUT P1(J)
470 IF P1(J)<0 THEN PRINTSU$;:G
    OTO 460
480 IF V1(J)=0 AND X1(J)=0 AND
    P1(J)=0 THEN 410
490 PRINTGI$GI$
500 PRINTRV$A2$(J)":"RO$Z3$Z7$;
    :INPUT V2(J)
510 IF V2(J)<0 THEN PRINTSU$;:G
    OTO 500
520 PRINTRV$A2$(J)":"RO$;Z4$Z7$
    ;:INPUT X2(J)
530 IF X2(J)<0 THEN PRINTSU$;:G
    OTO 520
540 PRINTRV$A2$(J)":"RO$;Z5$Z7$
    ;:INPUT P2(J)
550 IF P2(J)<0 THEN PRINTSU$;:G
    OTO 540
560 IF V2(J)=0 AND X2(J)=0 AND
    P2(J)=0 THEN PRINTSU$SU$SU$
    ;:GOTO 500
570 GOSUB 1780:IF RP=1 THEN 410
580 NEXT
590 :
600 REM
610 REM CALCOLO PERCENTUALI
620 REM
630 :
640 FOR J=1 TO NR
650 S1=V1(J)*P2(J):SX=X1(J)*X2(
    J):S2=P1(J)*V2(J)
660 IF S1+SX+S2=0 THEN S1(J)=33
    :SX(J)=33:S2(J)=34:GOTO 690
670 FR=100/(S1+SX+S2)
680 S1(J)=INT(S1*FR):SX(J)=INT(
    SX*FR):S2(J)=100-S1(J)-SX(J
    )
690 IF RT=1 THEN RETURN
700 NEXT
710 :
720 REM
730 REM STAMPA PERCENTUALI
740 REM
750 :
760 IF SQ=1 THEN TA=17
770 PRINTCL$ TAB(TA)X1$X2$X2$X3
    $X2$X2$X2$X3$X2$X2$X2$X3$X2
    $X2$X2$X4$
780 PRINT TAB(TA)X6$"NR"X6$" 1
    "X6$" X "X6$" 2 "X6$
790 PRINT TAB(TA)X7$X2$X2$X5$X2
    $X2$X2$X5$X2$X2$X2$X5$X2$X2
    $X2$X8$
800 X=0:FOR J=1 TO NR
810 PRINT TAB(TA)X6$X9$X9$X6$X9
    $X9$X9$X6$X9$X9$X9$X6$X9$X9
    $X9$X6$
820 X=X+1:IF X>2 AND J<10 THEN
    840
830 NEXT:GOTO 850
840 PRINT TAB(TA)X7$X2$X2$X5$X2
    $X2$X2$X5$X2$X2$X2$X5$X2$X2
    $X2$X8$:X=0:GOTO 830
850 PRINT TAB(TA)XA$X2$X2$XB$X2
    $X2$X2$XB$X2$X2$X2$XB$X2$X2
    $X2$XC$
860 IF SQ=0 THEN 930
870 PRINTHO$GI$GI$X1$XD$X3$XD$X
    4$
880 X=0:FOR J=1 TO NR
890 PRINTX6$SPC(7)X6$SPC(7)X6$:
    X=X+1:IF X>2 AND J<10 THEN
    910
900 NEXT:GOTO 920
910 PRINTX7$XD$X5$XD$X8$:X=0:GO
    TO 900
920 PRINTXA$XD$XB$XD$XC$
930 PRINTHO$GI$GI$;
940 PRINT TAB(33)X1$X2$X2$X2$X4
    $:X=0:FOR J=1 TO NR
950 PRINT TAB(33)X6$X9$X9$X9$X6
    $:X=X+1:IF X>2 AND J<10 THE
    N 970
960 NEXT:GOTO 980
970 PRINT TAB(33)X7$X2$X2$X2$X8
    $:X=0:GOTO 960
980 PRINT TAB(33)XA$X2$X2$X2$XC
    $
990 IF SQ=0 THEN 1040
1000 PRINTHG$;:X=0:FOR J=1 TO NR
1010 PRINT"[RIGHT]"LEFT$(A1$(J),
     7) TAB(9)LEFT$(A2$(J),7):X=
     X+1:IF X>2 AND J<10 THEN 10
     30
```

AVVISO

Per i Signori Negozianti e Operatori Settore
prossima apertura del Centro
Computer

```
1020 NEXT:GOTO 1040
1030 PRINTGI$;:X=0:GOTO 1020
1040 PRINTHG$;:X=0:FOR J=1 TO NR
1050 PRINT TAB(TA+4-LEN(STR$(J))
     );MID$(STR$(J),2,2):X=X+1:I
     F X>2 AND J<10 THEN 1070
1060 NEXT:GOTO 1080
1070 PRINTGI$;:X=0:GOTO 1060
1080 PRINTHG$;:X=0:TA=TA+8:FOR J
     =1 TO NR
1090 PRINT TAB(TA-LEN(STR$(S1(J)
     )));MID$(STR$(S1(J)),2,3);
1100 PRINT TAB(TA+4-LEN(STR$(SX(
     J))));MID$(STR$(SX(J)),2,3)
     ;
1110 PRINT TAB(TA+8-LEN(STR$(S2(
     J))));MID$(STR$(S2(J)),2,3)
1120 X=X+1:IF X>2 AND J<10 THEN
     1140
1130 NEXT:GOTO 1150
1140 PRINTGI$;:X=0:GOTO 1130
1150 PRINTHG$;:X=0:FOR J=1 TO NR
     :A$=""
1160 IF S1(J)>22 THEN A$="1"
1170 IF SX(J)>22 THEN A$=A$+"X"
1180 IF S2(J)>22 THEN A$=A$+"2"
1190 IF A$="1X2" THEN 1220
1200 IF LEN(A$)=2 THEN A$=" "+A$
     :GOTO 1220
1210 A$="  "+A$
1220 PRINT TAB(37-LEN(A$))RV$A$:
     X=X+1:IF X>2 AND J<10 THEN
     1260
1230 NEXT
1240 PRINTGI$GI$RV$"'F1': RESTAR
     T            'F3': MODIFICH
     E"
1250 GOTO 1270
1260 PRINTGI$;:X=0:GOTO 1230
1270 GET A$:IF A$=CHR$(133) THEN
      RUN
1280 IF A$=CHR$(134) THEN 1300
1290 GOTO 1270
1300 PRINTSU$SU$;:FOR J=1 TO 40:
     PRINTCHR$(32);:NEXT:PRINT
1310 POKE 214,21:SYS58640
1320 NI=0:INPUT "NUMERO INCONTRO
     ";NI:IF NI<0 OR NI>13 THEN
     1310
1330 IF NI=0 THEN POKE 214,19:SY
     S58640:GOTO 1240
1340 PRINTCL$"INCONTRO NR:";NI
1350 PRINTGI$ TAB(23)"VINTE NULL
     E PERSE"
1360 IF SQ=1 THEN PRINTA1$(NI);:
     GOTO 1380
1370 PRINT"SQUADRA OSPITANTE";
1380 PRINT TAB(26-LEN(STR$(V1(NI
     ))))V1(NI) TAB(32-LEN(STR$(
     X1(NI))))X1(NI);
1390 PRINT TAB(38-LEN(STR$(P1(NI
     ))))P1(NI)
1400 PRINTGI$;:IF SQ=1 THEN PRIN
     TA2$(NI);:GOTO 1420
1410 PRINT"SQUADRA OSPITE";
1420 PRINT TAB(26-LEN(STR$(V2(NI
     ))))V2(NI) TAB(32-LEN(STR$(
     X2(NI))))X2(NI);
1430 PRINT TAB(38-LEN(STR$(P2(NI
     ))))P2(NI)
1440 PRINTGI$GI$GI$"COSA MODIFIC
     HI?"
1450 PRINTGI$"1)NOMI SQUADRE":PR
     INT"2)DATI SQUADRE":PRINT"3
     )NULLA"
1460 GET A$:IF A$="1" THEN 1500
1470 IF A$="2" THEN 1570
1480 IF A$="3" THEN TA=0:GOTO 76
     0
1490 GOTO 1460
1500 IF SQ=0 THEN 1460
1510 A$="":PRINTCL$ TAB(19)A1$(N
     I):PRINTHO$;:INPUT "SQUADRA
      OSPITANTE";A$
1520 IF A$="" THEN 1510
1530 A1$(NI)=A$
1540 A$="":PRINT TAB(15)A2$(NI):
     PRINTSU$;:INPUT "SUADRA OSP
     ITE";A$
1550 IF A$="" THEN PRINTSU$;:GOT
     O 1540
1560 A2$(NI)=A$:GOTO 1340
1570 PRINTCL$;:IF SQ=1 THEN PRIN
     TA1$(NI):GOTO 1590
1580 PRINT"SQUADRA OSPITANTE"
1590 PRINT TAB(14)V1(NI):PRINTSU
     $;:A=0
```

```
1600 INPUT "VINTE IN CASA";A:IF
     A<0 THEN PRINTSU$;:GOTO 160
     0
1610 V1(NI)=A:A=0:PRINT TAB(16)X
     1(NI):PRINTSU$;
1620 INPUT "PAREGGI IN CASA";A:I
     F A<0 THEN PRINTSU$;:GOTO 1
     620
1630 X1(NI)=A:A=0:PRINT TAB(16)P
     1(NI):PRINTSU$;
1640 INPUT "PERDITE IN CASA";A:I
     F A<0 THEN PRINTSU$;:GOTO 1
     640
1650 P1(NI)=A
1660 IF V1(NI)=0 AND X1(NI)=0 AN
     D P1(NI)=0 THEN 1570
1670 PRINTCL$;:IF SQ=1 THEN PRIN
     TA2$(NI):GOTO 1690
1680 PRINT"SQUADRA OSPITE"
1690 PRINT TAB(17)V2(NI):PRINTSU
     $;:A=0
1700 INPUT "VINTE FUORI CASA";A:
     IF A<0 THEN PRINTSU$;:GOTO
     1700
1710 V2(NI)=A:A=0:PRINT TAB(19)X
     2(NI):PRINTSU$;
1720 INPUT "PAREGGI FUORI CASA";
     A:IF A<0 THEN PRINTSU$;:GOT
     O 1720
1730 X2(NI)=A:A=0:PRINT TAB(19)P
     2(NI):PRINTSU$;
1740 INPUT "PERDITE FUORI CASA";
     A:IF A<0 THEN PRINTSU$;:GOT
     O 1740
1750 P2(NI)=A
1760 IF V2(NI)=0 AND X2(NI)=0 AN
     D P2(NI)=0 THEN 1670
1770 RT=1:J=NI:GOSUB 650:RT=0:GO
     TO 1340
1780 PRINTGI$GI$"CONFERMI?":RP=0
1790 GET A$:IF A$="S" THEN RETUR
     N
1800 IF A$="N" THEN RP=1:RETURN
1810 GOTO 1790
1820 END
```

```
100 REM STAMPA COLONNE DEL FILE
110 REM GENERATO CON TOT-13
120 REM (C.C.C. N.35)
130 REM BY ANTONIO PASTORELLI
140 :
150 PRINTCHR$(147) TAB(11)"TOT"
160 PRINTCHR$(19);:INPUT "NOME
    FILE";A$:IF A$="" THEN 160
170 PRINTCHR$(17)"IL FILE "A$"
    E' REGISTRATO SU:"
180 PRINTCHR$(17)"  DISCO"
190 PRINTCHR$(145);:INPUT DV$:I
    F DV$="DISCO" THEN 220
200 IF DV$="NASTRO" THEN 240
210 GOTO 190
220 OPEN 1,8,15:CLOSE 1:IF ST<>
    0 THEN PRINTCHR$(147)"ACCEN
    DI IL DRIVE.":STOP
230 OPEN 1,8,12,A$+",S,R":GOTO
    250
240 OPEN 1,1,0,A$
250 OPEN 2,4,15:CLOSE 2:IF ST<>
    0 THEN PRINTCHR$(147)"ACCEN
    DI LA STAMPANTE.":GOTO 350
260 OPEN 2,4,7
270 PRINTCHR$(147)"'S' PER INTE
    RROMPERE LA STAMPA."
280 INPUT#1,NR:X=0:FOR J=1 TO N
    R:PRINT#2, TAB(7-LEN(STR$(J
    )))CHR$(18)JCHR$(146);
290 FOR X=1 TO 13:GOSUB 370
300 INPUT#1,A:IF A=1 THEN A$="1
    "
310 IF A=2 THEN A$="X"
320 IF A=3 THEN A$="2"
330 PRINT#2,A$;CHR$(32);:NEXT:P
    RINT#2:NEXT
340 PRINT#2:CLOSE 2
350 CLOSE 1
360 END
370 GET Z$:IF Z$="S" THEN 390
380 RETURN
390 PRINTCHR$(147)"PREMI UN TAS
    TO PER CONTINUARE ..."
400 PRINT"O 'F' PER FINE STAMPA
    ."
410 GET Z$:IF Z$="F" THEN 340
420 IF Z$<>"" THEN PRINTCHR$(14
    7)"'S' PER INTERROMPERE LA
    STAMPA.":RETURN
430 GOTO 410
440 END
```

Un anno di grafica

Una nuova iniziativa che sicuramente sarà apprezzata dai nostri lettori più affezionati; ma anche dai nuovi utenti Commodore

di Alessandro de Simone

"Dove è possibile trovare informazioni su un determinato argomento?"

Chi non si è mai posto questa domanda scagli il primo computer!

Poichè, anche noi della redazione, facciamo ormai fatica a rintracciare i titoli ed i programmi pubblicati sulla nostra rivista, ecco che abbiamo deciso di pubblicare, ogni mese, un elenco ragionato degli argomenti affrontati di recente su Commodore Computer Club.

In queste pagine troverete i titoli di articoli e programmi riguardanti solo la grafica computerizzta, pubblicati nell'arco degli ultimi dodici mesi: da ottobre '86 (n.35) fino a settembre '87 (n.45).

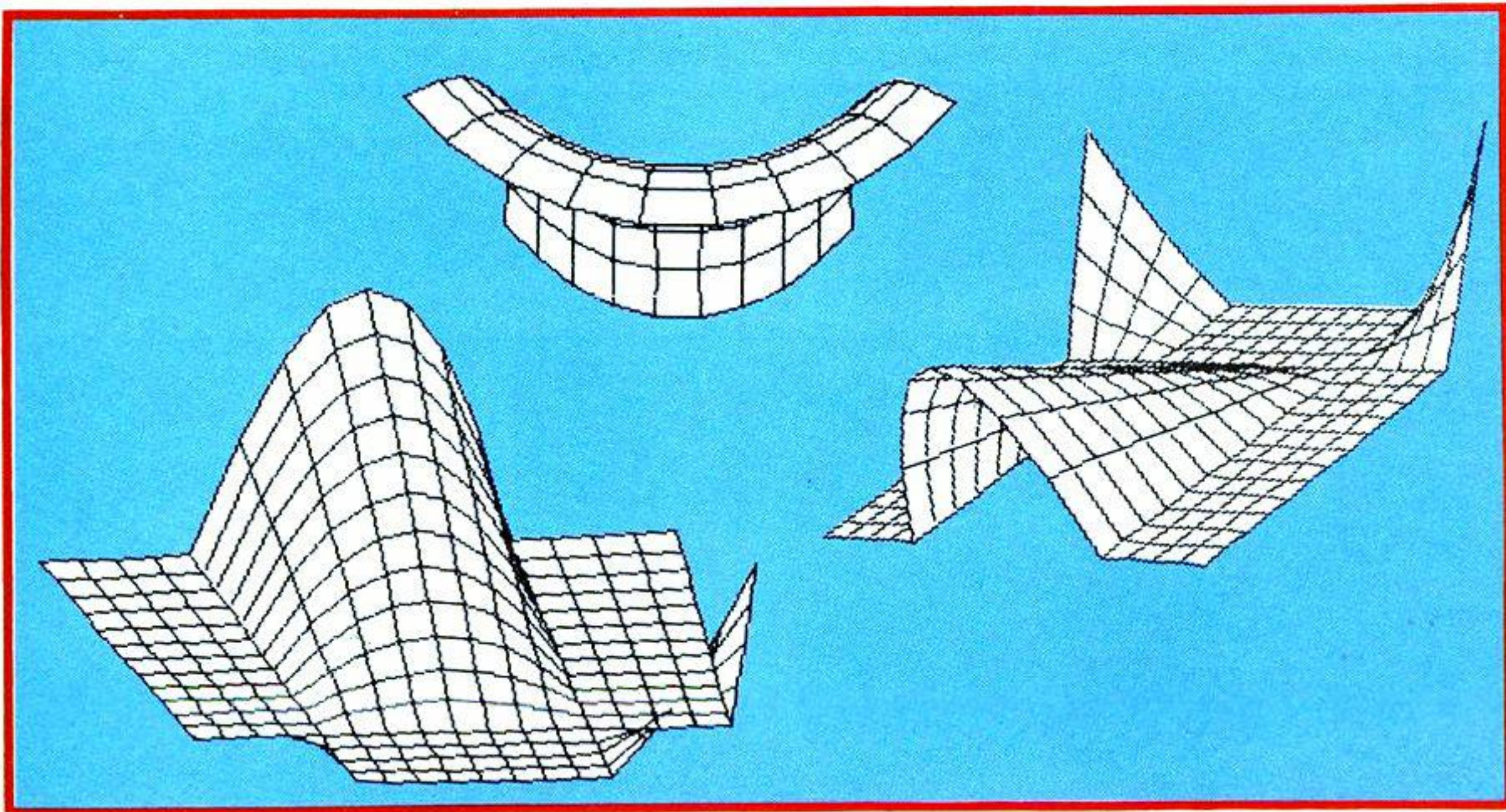
Fra parentesi è riportato il numero del fascicolo, mentre un breve riassunto indica succintamente il contenuto dell'articolo stesso.

I numeri di Directory, che contengono gli eventuali file programma, si possono determinare tenendo conto che al n.35 di C.C.C. era associato il n.1 di "Directory".

Naturalmente promettiamo di pubblicare, mese per mese, l'elenco aggiornato relativo agli argomenti più disparati.

(35) Graphic expander C/128 in 80 colonne
Recensione che descrive la sintassi dei comandi aggiuntivi in Basic relativi al software disponibile su disco presso la Systems Editoriale.

(35) Tutto sugli sprite
Un articolo didattico per sapere tutto ciò che desiderate sulla gestione di uno sprite.

(35) Cancella finestre schermo
Routine Basic dell'enciclopedia che consente di cancellare una porzione rettangolare del video di un C/64.

(35) Lampeggio righe schermo
Routine Basic dell'enciclopedia che rappresenta una delle numerose versioni disponibili per far lampeggiare una riga di testo del video di un C/64.

(36) Conversione di coordinate
Routine Basic dell'enciclopedia che converte coordinate cartesiane in polari e viceversa.

(36) Definizione Sprite
Routine l.m. dell'enciclopedia che consente di simulare, in Basic, la tecnica di definizione di uno sprite propria del Simon's Basic.

(36) Sprite Tool
Routine l.m. dell'enciclopedia che consente di individuare tutti i parametri relativi ad un determinato sprite mediante una semplice Sys dotata di ben otto parametri.

(36) Collisione
Routine l.m. dell'enciclopedia che consente di determinare facilmente l'eventuale collisione tra due sprite.

(37) Scroll
Quattro routine l.m. dell'enciclopedia che riproducono lo scrolling del video (in modo testo) nelle relative direzioni possibili.

(37) Scritte rotanti
Routine Basic dell'enciclopedia che visualizza sullo schermo del C/64 scritte scorrevoli ad una velocità programmabile.

(37) Plot in media risoluzione
Routine Basic dell'enciclopedia che consente di adattare lo schermo in modo testo del C/64 in un'area grafica di 80x50 punti.

(38) Koala per C/16
Un breve programma in Basic che consente di sfruttare completamente le risorse grafiche del piccolo computer.

(38) Mini disegni con la tua stampante
Un divertente programma in Basic per creare simpatici disegni da riportare su carta.

(38) Il primo giorno di scuola
Alcuni programmi grafici dimostrativi per il C/64 dotato di emulatore Gw-Basic.

(38) Draw in bassa risoluzione
Routine Basic dell'enciclopedia per tracciare semplici disegni sullo schermo del C/64.

(38) Due programmi... ipnotici!
Richiedono le routine di Toma questi due brevissimi programmi per C/64 che generano meravigliosi effetti grafici programmabili.

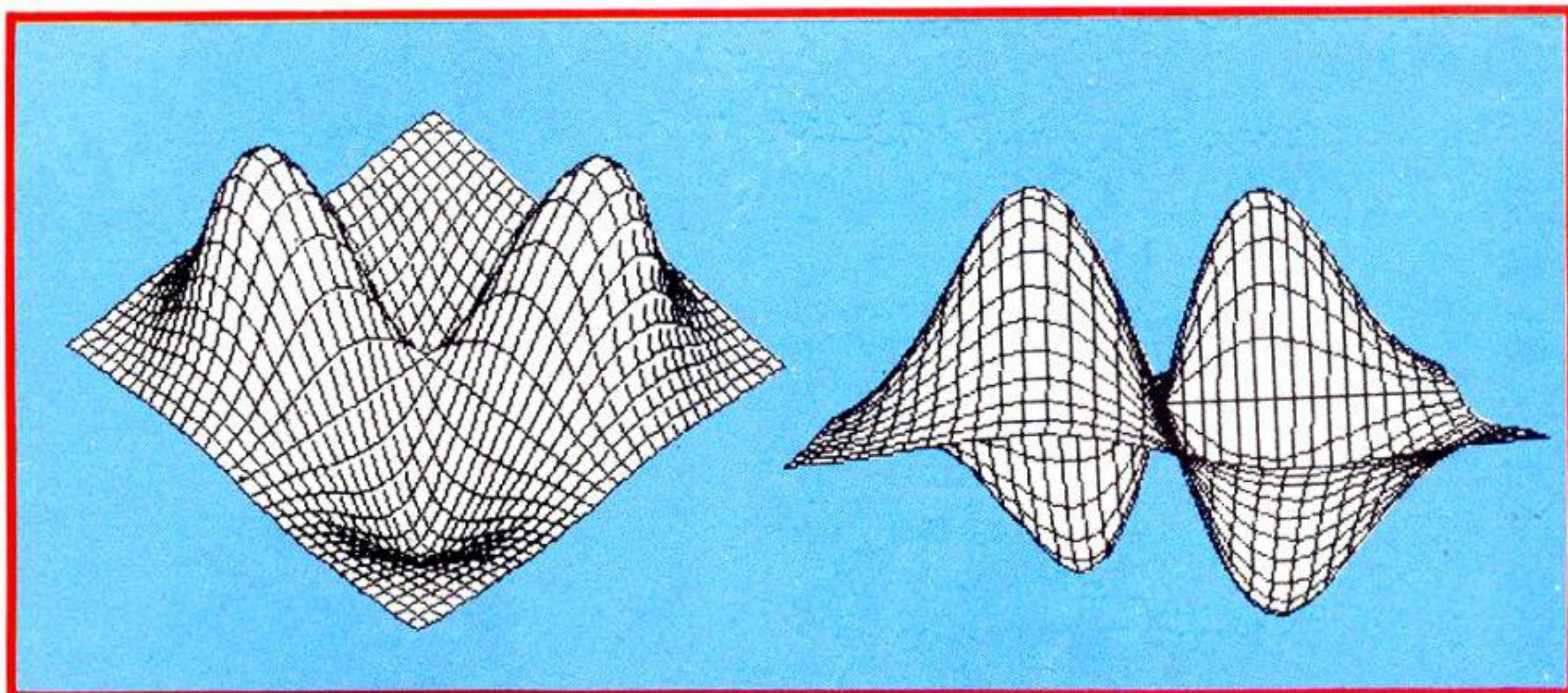
(39) Il giardino elettronico
Un brevissimo programma grafico, che richiede le routine di Toma, per la generazione di meravigliosi disegni pseudo-casuali sullo schermo del C/64.

(39) Sprite scanner
Routine Basic dell'enciclopedia che permette di esaminare la memoria alla ricerca dell'area RAM che contiene le informazioni relative ad un certo sprite.

(39) Una tavolozza per il tuo C/128
Un programma grafico scritto in Basic 7.0 che consente di eseguire disegni sullo schermo del potente computer in alta risoluzione.

(39) Riempi schermo
Routine l.m. dell'enciclopedia che riempe istantaneamente lo schermo del C/64 con un carattere precedentemente indicato.

(40) Un'anima(zione) per l'Amiga
Recensione dei programmi "Aegis Animator" e "Aegis Images" per Amiga.

(40) Page set/reset
Routine LM dell'enciclopedia per selezionare la pagina testo o la pagina grafica.

(39) Caleidoscopio
Un semplice programma, per C/64, per la generazione di suggestive immagini simmetriche e casuali.

(39) Curve e simmetrie
Un brevissimo programma grafico, per C/16, Plus/4 e C/128, che serve a generare bellissime immagini simmetriche pseudocasuali.

(39) Accensione sprite
Routine Basic dell'enciclopedia che provvede alla gestione completa delle operazioni elementari che riguardano uno sprite.

(39) Movimento sprite
Routine Basic dell'enciclopedia che consente di visualizzare la posizione attuale di un qualsiasi sprite.

(40) Page clear
Routine LM dell'enciclopedia per cancellare la pagina grafica.

(40) Page color
Routine LM dell'enciclopedia per accendere o spegnere un singolo punto nella pagina grafica.

(41) Hopalong: matematica e fantasia
Un'applicazione matematico-grafica per plottare meravigliose immagini pseudo-casuali con le routine di Toma.

(41) Grafica in Gw-Basic
Un breve programma che chiama in causa numerosissime istruzioni grafiche del simulatore Gw-Basic.

(41) Koala printer
Come riportare su stampante un disegno in alta risoluzione eseguito con un C/16 oppure con un Plus/4.

(41) Grandezze analogiche
Routine Basic dell'enciclopedia che utilizza la posizione reciproca di più sprite per visualizzare grandezze proporzionali.

(42) Rotazione di solidi in Hi-Res
Un'applicazione grafica di eccezionale interesse per coloro che utilizzano le routine di Toma con il C/64.

(42) Un pollo a testa, anzi nessuno
Introduzione alla statistica mediante un programma che richiede il preventivo caricamento del simulatore Gw-Basic.

(43) Quando uno schermo non basta
Un breve programma in Basic e l.m. (con disassemblato commentato) per tenere costantemente in memoria due schermate grafiche del C/128, richiamabili a piacere.

(43) Grafica in Basic 3.5 e 7.0
Due programmi, validi per C/16, Plus/4 e C/128, idonei per studiare le potenti istruzioni grafiche dei computer citati. Ed inoltre una routine di hard copy per lo schermo in alta risoluzione del C/128.

(44) Sprite mover
Una routine l.m. dell'enciclopedia (con disassemblato commentato) per spostare uno sprite sullo schermo del C/64 e rilevare eventuali collisioni.

(44) H/Copy Hi-Res
Una routine l.m. dell'enciclopedia (con disassemblato commentato) per riportare su carta una schermata in alta risoluzione del C/64.

(45) A grandi linee
Un programma per capire in che modo il C/64 gestisce la mappa di memoria Rom in cui sono allocati i caratteri di cui dispone.

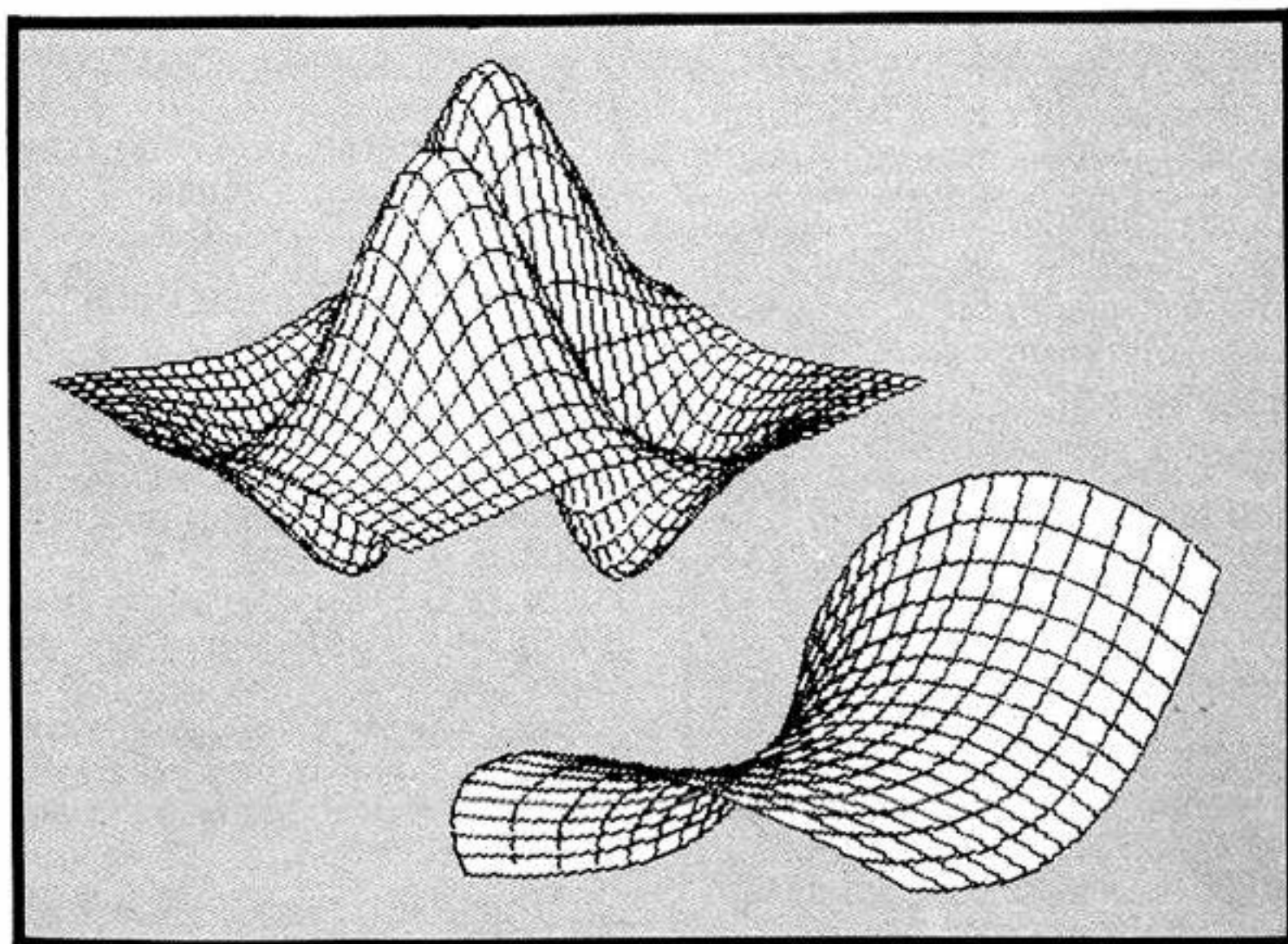
(45) Artisti si diventa
Un programma di mini-CAD, scritto in Basic, valido per C/16, Plus/4 e C/128.

(45) Espansione gratuita di memoria Ram
Come disporre di parecchie centinaia di byte in più ricorrendo ad un semplice trucco di programmazione sul C/16.

(45) Il peso della velocità
Inserto didattico che dimostra, con vari e semplici programmi, l'importanza della velocità di elaborazione.

(45) Incornicia risultati
Routine in Basic dell'enciclopedia che permette di rendere più efficace la visualizzazione di messaggi e valori numerici.

(45) Uno schermo in frantumi
Programma l.m, con disassemblato commentato, per utilizzare, con C/64 e drive, le schermate grafiche del programma Print Shop all scopo di ottenere effetti grafici interessanti.

(45) Programma grafico universale
Un programma in Basic che funziona allo stesso modo se digitato su C/16, Plus/4, Amiga, C/64+ Emulatore Gw-Basic e C/128!

Peek, Poke e Sys per C/64

Continua la rassegna dei comodi comandi Basic che consentono anche l'accesso a routine in linguaggio macchina

di Roberto Morassi

Per resettare (cioè riportare ai valori iniziali) i puntatori RAM del Sistema Operativo che siano stati alterati, non è sempre necessario il Reset completo (SYS 64738) che, oltretutto, effettua il NEW del programma in memoria.

Ecco una serie di SYS, che rimettono a posto specifici gruppi di puntatori, e che saranno apprezzate soprattutto dagli utilizzatori "evoluti" dal momento che possono essere usate solo in casi particolari:

SYS 65418
Resetta i puntatori di input / output compresi fra $0314 - 0333 (dec. 788 - 819), e cioè SAVE, LOAD, STOP, IRQ, NMI, BRK eccetera.

SYS 58451
Resetta i puntatori all'inizio della pagina 3 ($0300 - 030B, dec. 768 - 779), comprendenti il LIST, warm-start, messaggi di errore e decodifica parole Basic.

SYS 58370
Resetta i puntatori di inizio e fine della zona Basic ($2B-2C, $37-38; dec. 43-44, 55-56).

SYS 58303
Resetta il puntatore USR, quelli aritmetici, quelli di inizio e fine Basic, e la routine CHRGET ($73-8A, dec. 115 - 138). Non si può eseguire da programma, perchè genera un Syntax Error.

SYS 58792
Resetta il chip video VIC-II a $D000 - D02E, comprendente i puntatori degli sprites, i colori di schermo, i banchi di memoria, eccetera.

SYS 65409
Inizializza lo schermo, resetta i colori dei caratteri, fondo e bordo.

SYS 64760
Effettua un RESET abbreviato del sistema. Differisce dal SYS 64738 perchè non controlla la presenza del cartridge, non azzera le pagine di memoria 2-3, e non esegue il RAM-test.

SYS 64763
Come il precedente, però non resetta i puntatori di input - output.

Trucchi di programmazione in L.M.

Acquisizione dati semplificata

La sequenza di comandi...

```
JSR $AEFD
JSR $B79E
```

...che salta la virgola nel testo e poi raccoglie un dato numerico (a singolo byte) nel registro X, può essere sostituita con...

```
JSR $B7F1
```

...che esegue la stessa operazione con un risparmio di 3 bytes.

Rilocazione dei wedges

Supponiamo di voler inserire un wedge, ad esempio, nella routine di warm-start. Normalmente si usa un wedge -loader di questo tipo:

```
0334 LDA $3F     ;carica low-byte
0336 STA $0302 ;nel puntatore
0339 LDA $03     ;carica high-byte
033B STA $0303
033E RTS
033F inizio del wedge
..............
```

In questo caso, eseguendo SYS 820, il loader attiva il wedge inserendo il nuovo indirizzo nell'apposito puntatore. Se si vuole rilocare il tutto in altra zona di memoria, sorge un problema: anche ammesso che il wedge sia rilocabile (cioè privo di JSR e JMP interni) il loader non lo è, perchè si deve modificare il nuovo indirizzo da inserire nel puntatore.

Un semplice metodo per realizzare un loader rilocabile sfrutta il fatto che, quando si comanda un SYS, l'indirizzo di quest'ultimo rimane memorizzato in $14-15 (dec.#0-21) da cui può essere prelevato ed inserito nel puntatore previa aggiunta di un offset che lo riporta all'inizio del wedge. Esempio:

```
0334 LDA $14 ;legge low-byte dell'in-
              dirizzo del SYS
0336 CLC
0337 ADC #$10 ;aggiunge l'offset
              di inizio wedge
              (16 bytes)
0339 STA $0302 ;copia nel puntatore
033C LDA $15 ;legge high-byte
033E ADC #$00 ;incrementa se è
              in overflow
0340 STA $0303 ;copia nel puntatore
0343 RTS
0344 inizio del wedge
.......................
```

Sostituzione dei JMP interni

Se la rilocazione di un programma in l.m. è impedita dalla presenza di istruzioni JMP interne al programma stesso, si può in genere aggirare l'ostacolo sostituendo il JMP con un "branch" obbligato, purchè si conosca con precisione il valore di almeno uno dei flags di status. Ad esempio, il segmento...

```
C000 LDA #$00
C002 STA $FB
C004 JMP $C020
```

...si può sostituire con...

```
C000 LDA #$00
C002 STA $FB
C004 BEQ $C020
```

...rilocabile, perchè l'istruzione LDA #$00 lascia ovviamente settato il flag Z. In caso di incertezza si può ricorrere al trucco di resettare il flag di overflow (raramente usato) ed eseguire il branch con BVC:

```
C000  LDA $FC
C002  STA $FB
C004  CLV
C005  BVC $C020
```

Tutto ciò vale se la distanza fra i due indirizzi è inferiore a 128 bytes: se fosse superiore, si può sempre eseguire un "doppio branch", proteggendo il "branch" intermedio dal flusso del programma con un'istruzione BIT:

```
C000 LDA #$00
C002 STA $FB
C004 BEQ $C090
................
C08F BIT ;($2C: salta i due bytes
                bytes che seguono
C090  BEQ $C0D0
.................
```

Word processors e colori default

Nei word processors sono in genere previsti dei comandi per modificare il colore del fondo, del bordo e dei caratteri, così da adeguarli ai propri gusti nel caso questi non corrispondano ai valori di "default" inseriti nel programma.

Con qualche modifica si può però personalizzare il w.p. modificando questi valori in modo da avere i colori preferiti fin dall'inizio. Ecco come fare per due fra i w.p. più popolari: WORDPRO 3+ e EASY SCRIPT.

E' ovvio che le note che seguono potrebbero non andar bene per la copia di word processor in vostro possesso.

Wordpro 3+

Caricate il w.p. im memoria, ma senza dare il RUN. Supponiamo che i vostri colori preferiti siano indicati dai seguenti parametri:

F = colore del fondo (codice POKE)
B = colore del bordo (cod. POKE)
C = colore dei caratteri (cod. POKE)
H = colore dei caratteri (cod. CHR$).

Per esempio: caratteri blu su fondo grigio chiaro e bordo marrone:

F=15; B=9; C=6; H=31

Date i seguenti comandi in modo diretto, sostituendo ai parametri i colori prescelti:

```
POKE 2084,F: POKE 2079,B: POKE
2094,H: POKE 5953,C
```

Infine salvate il programma così modificato con altro nome (o su un altro disco).

Easy Script

Caricate il w.p. in memoria, senza dare il RUN. Date i seguenti comandi in modo diretto:

```
POKE 2525,F: POKE 2530,B
```

in cui F è il colore del fondo prescelto (cod. POKE) e B è il colore del bordo e dei caratteri (cod. POKE). Salvate il programma con un altro nome, o su un altro disco.

Notate che in Easy Script il bordo e i caratteri hanno inizialmente lo STESSO colore: se volete differenziarli, dovete farlo manualmente, dopo il Run, con Control-3.

Anti-New istantaneo

Se avete accidentalmente resettato il computer o dato il NEW, potete recuperare COMPLETAMENTE il programma Basic in memoria (quindi anche modificarlo, listarlo e dare il RUN) con questa miracolosa sequenza di comandi da digitare in forma abbreviata, in modo diretto e tutti di seguito, cioè senza spazi e con un unico RETURN.

```
POKE2050,1: SYS42291: POKE46,
PEEK(35)-(PEEK(781)>253):
POKE45,(PEEK(781)+2)AND255
```

Lampeggio durante il Get

Richiamando con un GOSUB questa subroutine (che sfrutta i registri di controllo del cursore e il contatore di tastiera) il programma si arresta, facendo lampeggiare il cursore sullo schermo: premendo un tasto, il cursore si spenge e il carattere del tasto viene assegnato alla variabile A$.

```
1000  POKE198,0:  POKE204,0:
WAIT198,1: WAIT207,1,1: GETA$:
POKE204,1: RETURN
```

SCHEDA TECNICA

Note divulgative per applicazioni varie.

Idoneo per il solo computer C/64.

Articolo consigliato ai lettori esperti

La Banca in Casa

Uno sportello del Credito Emiliano da installare comodamente sul vostro C/64

di Roberto Marigo

Il servizio bancario automatizzato ha ormai preso piede anche in Italia: quasi tutte le banche hanno adottato un metodo di cassa continua (Bancomat o equivalenti) e il gradimento del pubblico è comprovato dalla larga diffusione di sportelli che permettono il prelievo di denaro liquido a qualunque ora senza esser costretti a fare la fila davanti agli sportelli.

Il servizio offerto dal Credito Emiliano si inserisce nel filone dei servizi telematici dal momento che permette ai propri clienti di usufruire dei servizi bancari più richiesti, impartendo ordini da casa propria e in qualunque orario.

Come asserisce il chiaro ed esauriente manuale fornito a corredo del software, è possibile un nuovo modo di lavorare, che alleggerisce, da una parte, il lavoro degli impiegati e, d'altro canto, permette ai propri clienti una maggiore libertà ed una minore perdita di tempo.

E' consentito, tra l'altro, ordinare libretti di assegni, cambiare denaro in valuta estera o travellers cheques, e ritirarlo dopo un paio di giorni; oppure verificare il saldo del proprio conto corrente o l'esito di un assegno, senza alcun problema di orari.

Per usufruire di "BancainCasa", il servizio del Credito Emiliano di cui stiamo parlando, è necessario un Commodore 64, una qualsiasi memoria di massa (anche un semplice registratore a cassette!) e un modem del tipo Total Telecommunication della Telelearning System: il software necessario al collegamento viene fornito al momento della stipula del contratto con una qualsiasi filiale del Credito Emiliano.

Non sarà certo possibile a chiunque accedere alle informazioni sul vostro conto perchè il programma è protetto dall'accesso di intrusi in tre modi :

- *è presente un primo un codice, diverso per ciascun software assegnato a ogni utente, senza il quale il sistema stesso non gira.*
- *un secondo numero di codice individuale, anch'esso diverso per ciascun utente, controllato dal sistema centrale solo al momento del collegamento.*
- *un digit di controllo relativo ad ogni numero di conto corrente, controllato anch'esso dal computer centrale.*

Il dischetto provato non risulta protetto da copie, per cui l'utente può realizzare una copia di backup nel caso di perdita accidentale del programma o di smarrimento del dischetto.

Il collegamento avviene in tre fasi:

- la prima avviene localmente, prima di stabilire il collegamento via modem: il "colloquio" fra utente e C/64, in cui vengono formulate le richieste da inviare alla banca.
- la seconda fase è il contatto telefonico via modem con il computer del Credito Emiliano di Reggio Emilia.
- nella terza fase i dati di "ritorno" vengono presentati in un'unica schermata, con la possibilità di selezionarli singolarmente o di stamparli.

Il frazionamento presentato risulta più conveniente di un costante collegamento: una sessione di comunicazione telefonica richiederà, infatti, non più di 2 scatti, o di 4 se la comunicazione è effettuata in traffico interurbano e in ora di punta.

Se, invece, la scelte avvenissero in contatto costante con il computer centrale, la spesa telefonica salirebbe anche notevolmente (per selezionare cinque o sei scelte ci si mette in media 5-7 minuti, ma è molto di più se si richiede una o più stampe o se si vuole verificare il proprio conto).

Descriviamo ora il programma a noi pervenuto su dischetto:

Dopo l'introduzione e la schermata di presentazione, compaiono le brevi spiegazioni del funzionamento: con F1 si annulla un campo, con F3 si memorizza la scelta appena fatta, con F5 si elimina una schermata e si torna al menù cancellando la scelta precedentemente selezionata, con F7 si memorizza la scelta e poi si torna al menù principale, premendo CTRL-H si ottiene un aiuto in quasi ogni momento del programma, e con CTRL-P si può ottenere l'hardcopy del video.

A questo punto si inserisce il codice numerico e quello personale: ambedue sono resi non visibili, per garantire la segretezza se si fosse costretti ad operare in presenza di sguardi... indiscreti.

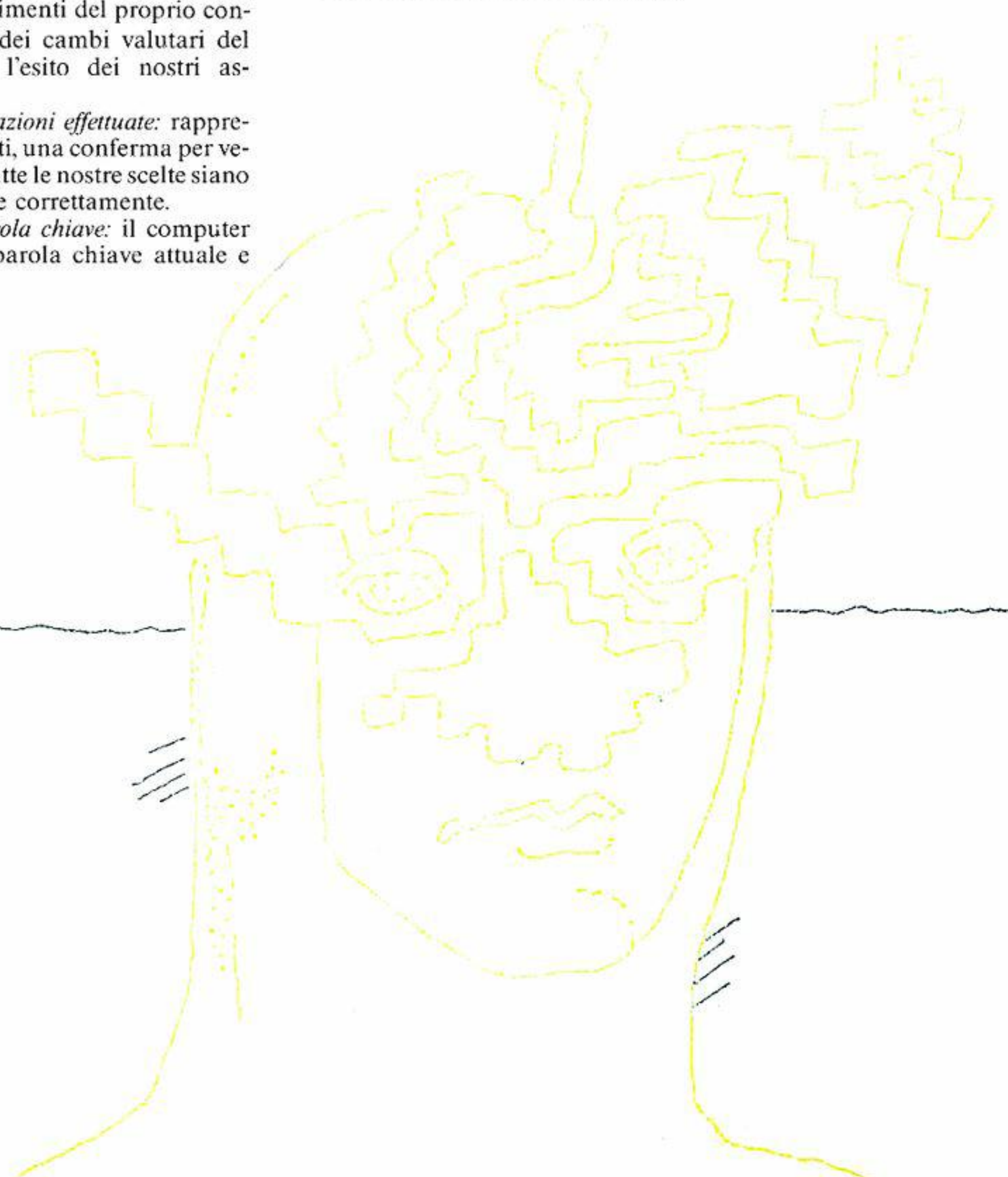
Solo se si verifica la corrispondenza, verrà caricata la seconda parte del programma.

Dal menù principale, che appare in seguito, si possono selezionare sei funzioni:

- *Informazioni:* si possono scegliere informazioni su consulenze finanziarie (gestione patrimoni mobiliari, fondi comuni di investimento), su prestiti personali (Bancofamiglia, prestito casa, prestito turistico ISEA), e su servizi (Prontobanca, Diners, Viacard, Pensione comoda, Utenze).

Le informazioni accessibili con questa opzione sono "incorporate" nel programma e non è quindi necessario stablire un collegamento via modem per ottenerle.
- *Comunicazioni:* questa, e quella successiva, sono le opzioni più importanti del pacchetto: limitandosi a digitare il numero di filiale e, ovviamente, il numero del conto, si può comunicare alla banca indicata la prenotazione di uno o più carnet di assegni, la prenotazione di assegni circolari, la prenotazione della valuta estera, l'invio di messaggi in formato libero e la richiesta di nuove funzioni tra quelle elencate nel programma stesso.
- *Richieste:* in questa opzione possono essere richiesti il saldo del C/C, la lista dei movimenti del proprio conto, il valore dei cambi valutari del momento e l'esito dei nostri assegni.
- *Elenco operazioni effettuate:* rappresenta, in effetti, una conferma per verificare che tutte le nostre scelte siano state accettate correttamente.
- *Cambio parola chiave:* il computer chiederà la parola chiave attuale e per due volte la nuova parola (per maggior sicurezza); al successivo collegamento la nuova parola chiave sarà riconosciuta valida dal computer centrale.
- *Collegamento:* dopo aver effettuato le scelte, il computer stabilirà autonomamente il collegamento con il centro dati del Credito Emiliano, chiedendo solo se l'utente si trova fuori dal distretto di Reggio Emilia e stabilire se formare, o meno, il prefisso teleselettivo. Se il collegamento non potesse avvenire (per linee occupate o altri problemi), verrà tentato un nuovo collegamento per tre volte di seguito, dopo di che il programma chiederà conferma prima di procedere con altri tentativi.

Una volta stabilito il collegamento, il computer avvertirà quando i dati sono in trasmissione o se la linea è disturbata: una volta concluso il collegamento il computer avvertirà l'utente della sua buona riuscita, presentando, con un menù, le richieste andate a buon fine: le risposte selezionate dall'utente, tramite la pressione del tasto corrispondente, appariranno sul video; sarà possibile a questo punto stamparle (CTRL-P), interrompere il lavoro con CTRL-Q, oppure iniziare una nuova sessione di lavoro con F3.

Il Credito Emiliano è una delle prime banche italiane a offrire un servizio del genere e garantisce continui aggiornamenti del programma, compresi nel prezzo del servizio stesso.

A che gioco giochiamo?

Quattro recensioni su altrettanti videogame per il meraviglioso computer Amiga

di Luigi Callegari

Se hai acquistato da poco un Amiga, vorrai certamente procurarti del software che sia in grado di esaltarne le straordinarie potenzialità.

Ecco, quindi, alcuni commenti su quattro giochi facilmente reperibili anche in Italia.

The Faery Tale
(configurazione minima: Amiga 512K)

Prodotto dalla MicroIllusions, si tratta di un classico gioco di arcade/adventure. La scena di gioco è molto ampia, tanto che lo schermo funge da finestra, scorrevole fluidamente, centrata sul nostro protagonista. Il gioco, secondo una caratteristica che si sta affermando come standard per Amiga, viene controllato interamente tramite mouse, agendo su appositi ideogrammi visualizzati costantemente sulla sezione inferiore del video.

Il programma è fornito in elegante confezione (il prezzo, non per niente, si aggira sulle centomila lire) con manuale molto artistico su carta tipo pergamena.

Inserito il dischetto alla richiesta del Workbench inizia una gradevole presentazione grafica e sonora con un libro (con tanto di figure) che, come in un cartone animato, si sfoglia da solo.

I nostri eroi sono tre fratelli, corrispondenti nella pratica a tre sagome leggermente diverse per abbigliamento e capigliatura, ciascuno dotato di varie "vite", donate da una fatina. La loro missione consiste nel rintracciare il talismano che protegge il villaggio, da dove parte il gioco, dalle forze del Male.

Ai fini di protezione anti-pirati, vengono quindi richieste tre parole d'ordine, leggibili sul manuale originale; poi inizia il gioco vero e proprio con la selezione di alcuni menu, ciascuno dotato di sottomenù. Vediamo a che servono:

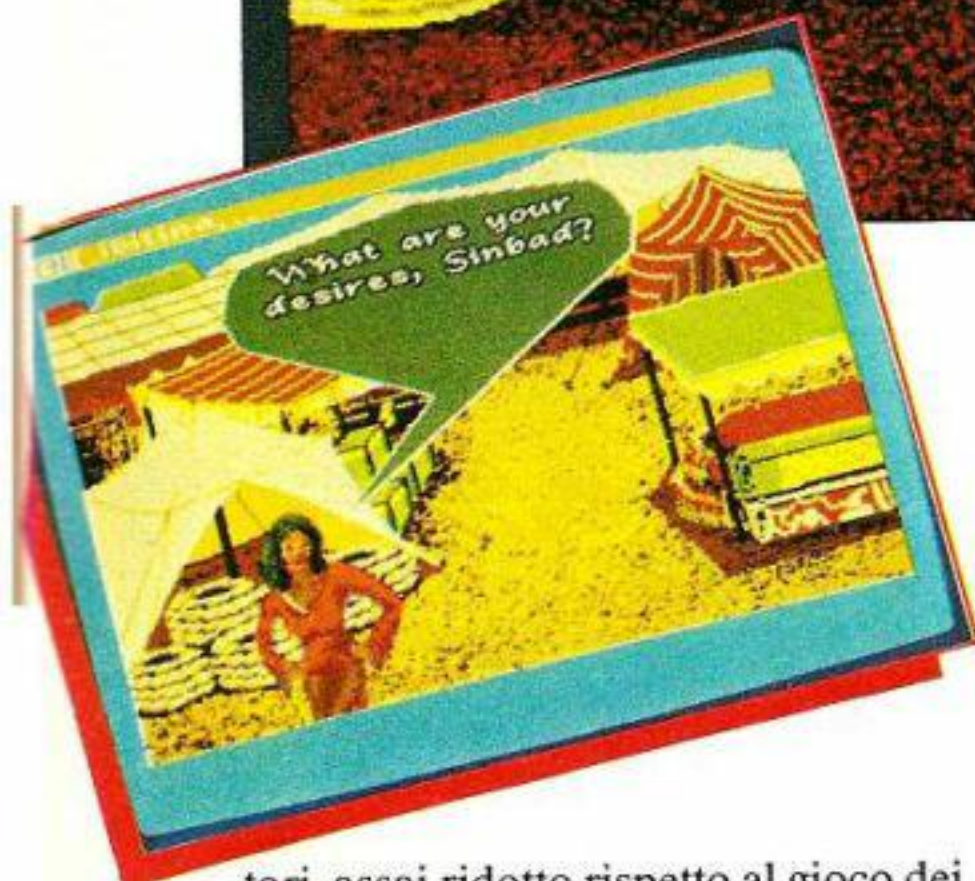
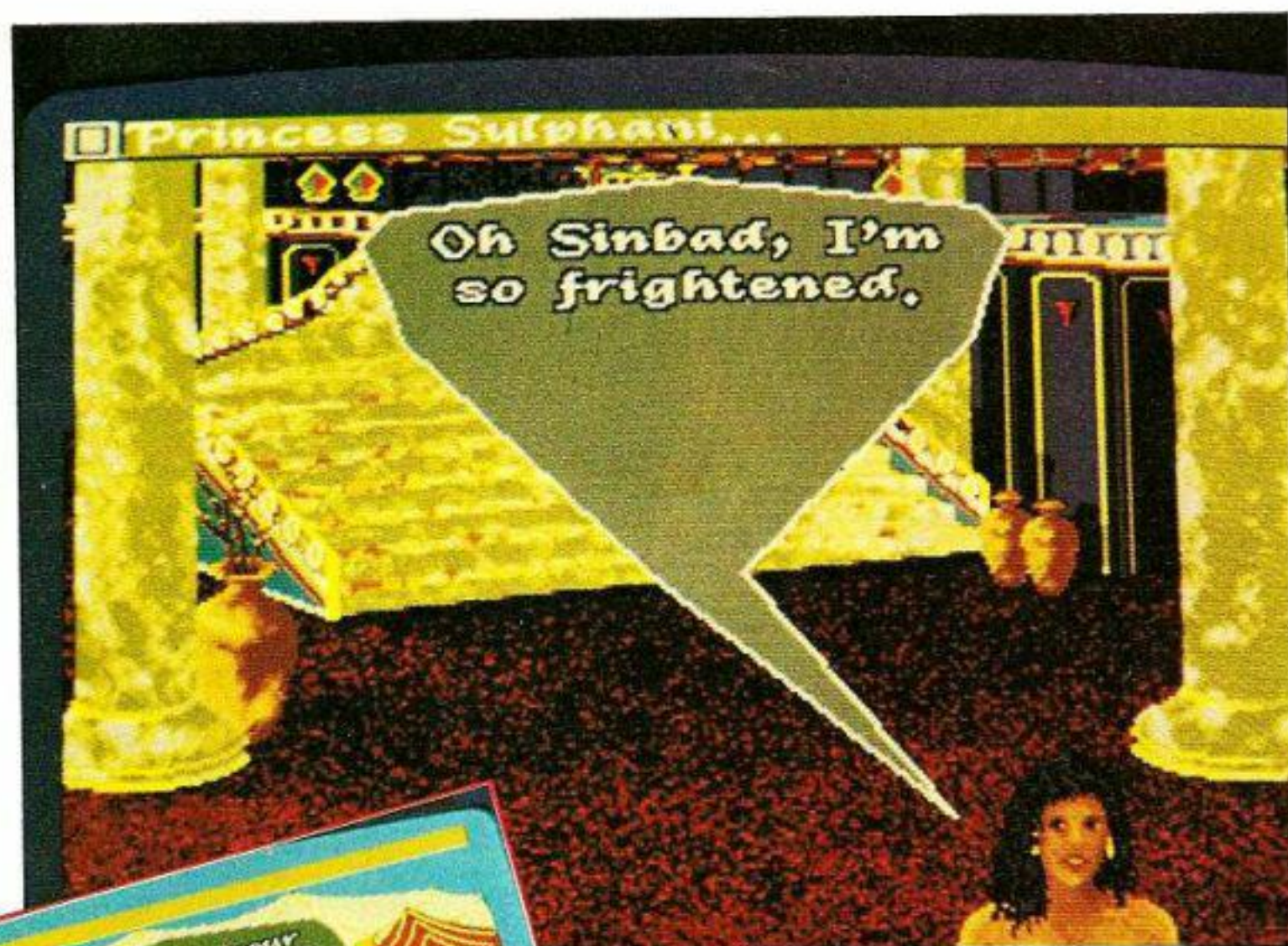
ITEMS: Per prendere, usare, dare oggetti od ottenere la lista di quelli in nostro possesso.

TALK: Per colloquiare, proprio come in un gioco di avventura sofisticato, con le altre creature del Mondo Fatato. Le sotto-opzioni permettono di parlare (SAY), urlare (YELL) e richiedere (ASK).

GAME: Permette di gestire il programma, terminandolo (QUIT), sospendendolo (PAUSE) oppure eliminando musica ed effetti sonori.

BUY: Permette l'acquisto di cibo, frecce, spada, totem, mazza, l'arco o le pozioni magiche. Il denaro, speso quello in dotazione, si può reperire nel gioco facendo bene attezione ai luoghi che si percorrono.

MAGIC: Permette di sfruttare gli oggetti magici trovati nel corso della partita. Ovviamente non sveliamo che cosa sono e come si usano, visto che è una delle parti più interessanti del gioco.

La scenografia è coloratissima e ben rifinita. Si pensi che, quando si rende necessario, la porzione di scena viene caricata da dischetto (in un batter di ciglio); evidentemente 500K di memoria centrale non bastavano e si sono sfruttati anche gli 880K del dischetto e la velocità dell'hardware di

tori, assai ridotto rispetto al gioco dei cavalieri, ha certamente tentato di renderlo più vario e sofisticato, aiutato da una maggiore esperienza accumulata dai primi pionieri.

Ne risulta un misto di avventura, azione, strategia, pagine grafiche raffinate, animazioni velocissime, effetti sonori e musiche suggestive, in grado di fissare molti nuovi standard di riferimento per i futuri videogiochi Amiga, che finalmente incominciano a sfruttare la macchina come si deve e non un C/64 con più memoria e colori (ogni riferimento a giochi come "Winter Games" e "One on One" è puramente voluto...).

La trama del gioco si basa, molto liberamente, sulla raccolta "The Arabian Nights", quindi sulle peripezie del marinaio Sinbad da Damaron alle prese con un annoso e pericoloso problema. La principessa Sylphani, figlia del Califfo di Damaron, ha visto suo padre trasformato in falcone da un misterioso incantesimo. Per guarirlo occorre spezzare l'incantesimo prima che sia troppo tardi: una specie di gigantesca bussola segna il trascorrere del tempo, all'esaurimento del quale il gioco termina ed il sultano dovrà rassegnarsi a svolazzare tra i rami per il resto dei suoi giorni. Il modo di rompere l'incantesimo deve scoprirlo Sinbad, cioè voi.

Durante il gioco si può colloquiare con sei personaggi, che possono sia entrare a far parte della nostra compagnia (e quindi sempre disponibili), che essere richiamati ogni volta che serve. Il dialogo avviene tramite fumetti (per l'interlocutore) e un certo numero di risposte selezionabili col mouse. Queste riducono il numero di varianti possibili (aumentate solo dall'umore del momento del nostro interlocutore), ma semplificano di molto il gioco rispetto ad un'avventura classica, tipo "The Pawn".

I personaggi con cui possiamo parlare sono: Sylphani, il fratello del Califfo Darun, la strega ex-moglie del califfo Libitina, lo stregone Shaman, la Zingara, il Genio della Lampada. Si noti che la personalità e la funzione di ciascuno di questi personaggi è tutta da scoprire, impiegando molto tempo nel corso di numerose partite. Ad esempio, la Zingara, se opportunatamente lusingata, può rivelare quasi sempre la posizione geografica ove rintracciare personaggi o nemici e può essere molto generosa con noi; Darun, se preso col dovuto rispetto ed evitando argomenti scottanti, può spiegarci i segreti per battere a duello o superare le varie disgrazie del gioco.

Le loro personalità spesso collidono: ad esempio sarà impossibile tenere nella nostra compagnia, per molto tempo insieme, Libitina e Sylphani: questa scapperà se non allontaniamo la prima, che però è utilissima come strega: ci toccherà andarla a cercare per tutto il globo!

Ciò che rende affascinante il gioco è che la maggior parte degli avvenimenti è da scoprire, dialogando con i vari personaggi ed andandosene a zonzo per il mondo a bordo del nostro Sabralus. Quando qualcuno ci dice di andare da qualche parte è necessario consultare, per mezzo di una lente di ingrandimento(!), la mappa animata del globo; in seguito dobbiamo, pazientemente, tracciare la rotta ed avventurarci tra tempeste ed insidie varie.

Il gioco si suddivide spesso in numerosi sotto-giochi: per vederli tutti è necessario completare parecchie partite, ciascuna con proprie caratteristiche arcade. Ad esempio, bisogna abbattere alcuni pterodattili a colpi di balestra (grafica in tempo reale, quasi tridimensionale), recuperare naufraghi dribblando gli scogli (in 3D), uccidere a colpi di fionda tre ciclopi, duellare con il Principe Nero, leoni e mitiche creature, risalire dal fondo di crepacci aperti da terremoti ed altre riposanti attività del genere. La buona grafica e la velocità di movimento sono veramente notevoli.

Dicevamo che vi sono più fasi strategiche; in una di queste Sinbad deve guidare, per mezzo di una mappa, gli eserciti del Califfo per combattere quelli del Principe Nero che vuole entrare nel palazzo. Quando poi si intraprendono viaggi per mare, bisogna decidere quante persone dell'equipaggio inviare per reclutare nuove leve; quando si approda su di una nuova terra, è necessario stabilire quanti personaggi portare con sè per

gestione! Lo scorrimento è fluidissimo, gli effetti sonori sono limitati, ma validi, specialmente durante le lotte con i numerosi nemici che si aggirano: scheletri, briganti eccetera.

L'idea del programma sembra valida, anche se semplice, così come il miscuglio tra avventura (che costringe a riflettere) e arcade (lotte a colpi di mouse, pardon, spada); il tutto condito dalle caratteristiche extra della grafica e del suono Amiga che, pur se sfruttati in piccola parte, permettono di surclassare i vecchi e gloriosi videogiochi del C/64.

Barbarian

(configurazione minima: Amiga 512K)

Si tratta di un classico gioco arcade, interamente controllato da mouse, dove si controlla un omaccione che deve combattere contro nemici ed insidie mortali per uccidere il malvagio mago Drax.

Nella parte inferiore del video sono presenti alcuni ideogrammi che consentono, al nostro omino, di svolgere varie azioni: muoversi avanti ed indietro (ovvero verso sinistra o verso destra sullo schermo bi-dimensionale), scendere o salire (quando sono presenti scale), sferrare colpi di spada, deporla e lanciarsi a lottare a mani nude, saltare ostacoli.

Pigiando il pulsante destro del topo si commuta un sotto menu, sempre visualizzato nella sezione inferiore del video, che permette di raccogliere e depositare oggetti, nonchè afferrare o riporre armi (secondo l'avversario che vogliamo affrontare). Inoltre sono visualizzate le lancie raccolte ed il numero di vite rimaste.

Il gioco procede attraversando vari schermi il cui scorrimento non è continuo ma avviene, in un colpo solo, quando viene raggiunto il bordo destro o sinistro del video. Inutile dire che la scenografia è estremamente rifinita e coloratissima, specie se la si confronta con quella riproducibile da un C/64. A proposito, chi avesse visto la versione per C/64 di questo stesso gioco, tenga conto che la versione Amiga è notevolmente diversa (ovviamente migliore sotto quasi tutti i punti di vista).

Qualche riserva, invece, è da esprimere sulle animazioni grafiche, pur se sembra evidente che Amiga riserva ancora molte difficoltà ai programmatori, come è ovvio per una macchina così complessa e nuova.

Infatti i movimenti del nostro barbaro non sono proprio il massimo della fluidità, così come quelli dei mostri suoi avversari.

Confrontato con i primi programmi per C/64, questo Barbarian è incomparabilmente superiore; se, poi, pensiamo agli enormi progressi fatti dai videogiochi per il C/64 nel tempo, pur essendo una macchina dalle caratteristiche hardware decisamente patetiche se confrontate con quelle di Amiga, non possiamo fare a meno di rilevare che, tra qualche tempo, i giochi saranno spaventosamente belli.

A proposito di grafica, piacevole è l'animazione che precede l'inzio del gioco ove si vede, come in un bel cartone animato, un barbaro che rompe una catena con la spada: il realismo è impressionante.

Validi anche gli effetti sonori, sebbene siano quasi sempre sincronizzati male con l'azione. Visto che il gioco è stato sicuramente scritto in buona parte in linguaggio C ed essendoci nota le difficoltà di gestione dei vari "task" in un ambiente multiprogrammabile come Amiga, non possiamo non perdonare i programmatori del videogame che, come già detto, sono anche "novellini" di questa complicatissima macchina; siamo certi, comunque, che si realizzerà di meglio con qualche sforzo in più.

Tutto sommato Barbarian è un bel gioco, specie se si considera l'ancora scarno panorama di videogiochi nel software Amiga.

Una nota: si consiglia vivamente l'acquisto dell'originale, disponibile ad esempio presso la Lago Snc (Via Napoleone, 16 - Como) non solo per i consueti motivi di ordine morale e pratico (elegantissima la confezione ed i gadgets) ma anche per il fatto che il dischetto è dotato di un'originale protezione hard/soft (che ovviamente non sveliamo) in grado di dare molti problemi ai pirati ed ai copiatori di dischetti.

Sinbad ed il Trono del Falcone

(configurazione minima: Amiga 512K, Kickstart 1.2, drive supplementare)

Si tratta di uno stupendo gioco, prodotto dalla Cinemaware e distribuito dalla Mindscape, per il quale il drive supplementare si rende praticamente indispensabile.

Chi possiede Amiga conosce "Defender of The Crown", della stessa casa, ed avrà apprezzato, giustamente, la pregevole grafica, ma non certamente la stucchevolezza del gioco.

Con Sinbad il team di programma-

ogni evenienza e quanti lasciare a guardia della nave per difenderla dai pirati.

Le cose, ovviamente, sono sempre più complicate di quanto possiamo immaginare: ad esempio, il numero di reclutati dipende anche dall'ostilità del territorio in cui muoviamo e si possono verificare perdite ingenti tra i marinai.

Ci sarebbero ancora moltissime cose da dire, soprattutto sulle innumerevoli pagine grafiche di scenografia che rendono il gioco sempre diverso ed affascinante. Crediamo, tuttavia, che basti affermare che si tratta di uno dei migliori giochi disponibili per Amiga, assolutamente imperdibile.

"Sinbad and the Throne of the Falcon" è distribuito dalla Mindscape Inc. 3444 Dundee Road, Northbrook, IL 60062, USA e costa $49.95 su due dischi.

Flight Simulator II

(configurazione minima: Amiga 512K)

E' una delle più famose simulazioni, già realizzata per la maggior parte dei computer: Apple, PC IBM, Atari e C/64.

Precisiamo subito che in circolazione esistono due versioni di questo programma: una, più vecchia, adatta a Kickstart 1.1 ed una, recentissima, per Amiga 500/2000 ed Amiga 1000 dotato di Kickstart 1.2. Le differenze sono praticamente inavvertibili, e consistono essenzialmente nella minore probabilità di mandare in "crash" l'elaboratore (con la nuova versione) ed in una nuova protezione del dischetto originale contro le copie pirata.

A chi non conosce il programma, possiamo dire che si tratta di una efficace simulazione di volo che fino a pochi anni fa era di competenza di elaboratori enormemente più costosi.

Il programma tiene conto di ben 47 parametri tipici di un vero volo in aereo. La presentazione è in grafica tridimensionale colorata ed estremamente realistica, indiscutibilmente superiore a quelle viste su qualunque altro microcomputer (e vi assicuriamo di averne viste tantissime). Evidentemente i programmatori hanno ben sfruttato le qualità superiori dell'hardware Amiga.

E' possibile pilotare uno di due velivoli: un Cessna 182, economico monomotore ad elica, ed un meno docile bireattore Gates Learjet. Non manca di certo la tradizionale opzione per simulare combattimenti aerei.

Il programma presenta costantemente ben sette menu del tipo Amiga/Intuition, selezionabili via mouse. Le opzioni sono numerosissime, ma ricordiamo: visione del gioco (quattro viste dalla cabina, vista dalla torre dell'aeroporto, da un altro aeroplano), zoom ottico, scelte metereologiche (stagioni, presenza forza e direzione vento, pioggia, nuvole), mappa di volo animata.

I principianti apprezzeranno molto il pilota automatico e l'opzione DEMO, nonchè il fatto che il tasto HELP può in ogni istante istruire su un comando qualunque. Sono inoltre presenti su dischetto alcune rotte dimostrative che permettono di vedere in azione il programma (consigliamo quella per visionare la Statua della Libertà). Inoltre è possibile salvare e ricaricare in ogni momento le situazioni di volo personali.

Mancano all'appello, purtroppo, i dati per atterrare sulla Piazza Rossa...

A proposito del dischetto, c'è da annotare quella che crediamo sia forse l'unica pecca del programma: quando si accede al floppy, il programma spesso viene momentaneamente "congelato". Cosa evitabile con una tecnica di programmazione più curata (diamine, Amiga lavora in multitasking!), come abbiamo visto in altri programmi commerciali.

Tra le caratteristiche "inedite" di questo programma vi è la possibilità di aprire più finestre contemporaneamente, tra cui la mappa, la visione laterale e lo zoom. Inoltre, udite udite, il programma è predisposto per il collegamento con un altro Amiga, permettendo così, a due giocatori, di volare contemporaneamente, ciascuno col suo elaboratore ma avendo anche nel proprio "mondo" l'aereo

controllato dall'amico.

Curatissima la strumentazione, che occupa quasi metà schermo, gli effetti sonori e la poderosa documentazione. Si badi che senza il manuale di 130 pagine è impossibile usare (e godere), se non in piccolissima parte, questo sofisticatissimo programma. Consigliamo vivamente l'acquisto del prodotto originale (circa centomila lire, in Italia).

Chiudiamo con un paragone che chiama in causa la versione per C/64. Programmi di questo genere impongono uno sfruttamento intensivo delle risorse della macchina, superiori alle "forze" del vecchio C/64, che spesso reagiva ai vari comandi con una lentezza esasperante. Amiga invece risponde immediatamente (anche troppo!) a qualunque comando, continuando a dare effetti sonori realistici ed una visione grafica nitida, colorata e suggestiva. Per non parlare del fatto che, grazie alla netta superiorità di memoria centrale (RAM) e di massa (dischi da 800K), questa versione permette ben cinque differenti aree di gioco e 120 (centoventi) aeroporti, con risoluzione grafica e velocità di gestione (grazie, Blitter!) tale da fare deprimere fino al suicidio i possessori dei vecchi elaboratori ad otto bit.

"Flight Simulator II" è prodotto dalla Sublogic Corporation, 713 Edgebrook Drive Champaign, IL61820, USA.

LA PERFEZIONE DIVENTA MITO

QUAD-MITO - 5 1/4" 96 TPI DS/QD
Floppy disk a quadrupla densità, disegnato per aumentare la capacità di registrazione sino a 780 kb per dischetto.
Velocità di registrazione 5800 BPI

MEGA-MITO - 5 1/4" 96 TPI HIGH DENSITY
Floppy ad alta densità, disegnato per drive da 1.2 MEG (AT e compatibili).
Velocità di registrazione 9650 BPI

MICRO-MITO - 3 1/2" 135 TPI DS/DD
Costruito per l'era dei disk drive da 3 1/2".
Velocità di registrazione 8100 BPI

le misure della perfezione

La *MICROFORUM MANUFACTURING INC.* è interessata all'ampliamento della propria rete distributiva. Per qualsiasi contatto scrivere anche in italiano.

944/A St. Claire Ave. West TORONTO, CANADA M6C 1C8
Tel. (416) 656-6406 - Tlx. 06-23303 MICROFORUM TOR
Telefax (416) 656-6368

A tutto disco.

*Finalmente, viste le numerose richieste, d'ora in poi ogni pubblicazione **Software Club** su cassetta sarà disponibile anche su dischetto da richiedersi, per corrispondenza, presso la redazione.*

***Software Club #11** (C/64-128)*
***Software Club #12** (C/64-128)*
***Software Club #13** (C/64-128)*
***Software Club #14** (C/64-128)*
***Software Club #15** (C/64-128)*
***Software Club #16** (C/64-128)*

***I Gialli Commodore 1-2-3** (C/64-128)*

***Charlie Deus** (C/64-128)*

***Ruby Rap** (C/64-128)*

***La voce III** (C/64-128)*

Il prezzo, per ognuna delle suddette pubblicazioni è di L. 12.000 più L. 3.000 per spese di spedizione.

Le richieste vanno indirizzate a:

Systems Editoriale
Viale Famagosta, 75
20142 MILANO
Tel. 02/8467348

Il pagamento può essere effettuato tramite assegno bancario o versamento sul c/c N. 37952207
Non è possibile inviare materiale in contrassegno nè contro invio di vaglia telegrafico.
Per ogni ordine, anche se per più dischetti, le spese rimangono fissate in L. 3.000

Sempre un passo avanti.

MS-DOS e Gw-Basic Versione 2

Tra breve in edicola la seconda versione del package che tanto successo ha incontrato presso i nostri lettori

di Alessandro de Simone

Finalmente è pronta, come avevamo promesso, la nuova realizzazione che consente di "trasformare" un semplice C/64 (dotato di disk drive) in un computer Ms-Dos Emulator.

Tra brevissimo tempo sarà disponibile in edicola la confezione contenente il dischetto su cui sono riportati i file che costituiscono il potente package operativo.

Le novità

La sezione MSDOS è stata completamente rifatta e sono disponibili quasi tutti i comandi (alcuni interni, altri esterni) presenti su un "vero" Ms-Dos compatibile. Chi se ne intende può giudicare da solo quanto i nuovi comandi siano interessanti:

Break, Chdir, Cd, Cls, Copy con, Copy, Date, Del, Erase, Dir, Diskcopy, Echo, Format, Mkdir, Path, Pause, Print, Prompt, Rem, Rename, Rmdir, Size, Sys, Time, Type, Ver, Vol.

Avete capito benissimo: è possibile creare subdirectory (fino a cinque livelli ciascuna) ed è possibile la procedura Path che consente di attivare, in successione, tutti i comandi dell'Emulatore MSDOS e perfino quelli creati dall'utente!

La sintassi è praticamente identica a quella "originale" Microsoft, i nomi dei file sono costituiti da otto caratteri (oltre a tre opzionali di esten-

sione) ed è attivabile la procedura Autoexec.

Perfino i messaggi di errore sono gli stessi ed identica risulta la loro interpretazione da parte del C/64.

Particolare cura è stata prestata per l'emulazione dei tasti funzione più importanti (F1, F2, F3, F4).

Chi ha lavorato qualche volta con il sistema operativo Ms-Dos può facilmente immaginare la validità del software proposto i cui autori sono Giancarlo Mariani e Fabio Sorgato, ben noti ai nostri lettori che apprezzano gli articoli sul Linguaggio Macchina.

La versione 2 Gw-Basic

La nuova versione del Basic è ancora più vicina a quella originale Gw-Basic.

Le modifiche più importanti sono state effettuate non solo per le nuove esigenze sulla gestione dei nomi (identica, stavolta, al "vero" Ms-Dos) e sulla possibilità di sfruttare totalmente le subdirectory; particolare attenzione, infatti, è stata anche prestata affinchè la sintassi dei file relativi risultasse identica, ed alcune istruzioni grafiche divenissero totalmente compatibili con l'edizione originale.

E' possibile, inoltre, registrare la pagina grafica e richiamarne altre memorizzate in precedenza.

La gestione dell'alta risoluzione, inutile dirlo, è quella tipica del C/64: 320x200 ed i comandi del colore sono rigorosamente identici a quelli disponibili su macchine professionali.

La velocità è certamente dignitosa e non farà rimpiangere l'utilizzo del C/64 in modo "normale".

Non dimentichiamo, infatti, che tutte le istruzioni normalmente disponibili sul C/64 rimangono attive anche con l'Emulatore Gw-Basic inserito nel computer.

Le istruzioni disponibili nella versione 2 del Gw-Basic sono le seguenti:

Append, Auto, Beep, Bload, Bsave, Buffer, Call, Chdir (path), Circle, Clear, Close, Cls, Color, Csrlin, Delete, Dirall, Disk, Draw, Edit, Eof, El, Else, Er, Erdev, Erdev$, Field, Files, Fix, Func1, Func2, Get#, Gwbasic, Hex$, Input$, Instr, Inkey$, &H, Key, Keyoff, Keyon, Kill, Lcopy, Line, Lineinput, Lineinput#, Llist, Locate, Load, Lprint, Lset, Merge, Name, Oct$, On (error, goto), Open, Output, Paint, Play, Point, Preset, Print, Pset, Put, Quit, Randomize, Renum, Restore (ln), Resume (next), Rget$, Rset, Reset, Resume, Run "name", Save, Screen, Sound, Space$, String$, Swap, Tape, Troff, Tron, Using, Version, Xor, Wend, While, Write, Write#

Non appena la nuovissima versione sarà commercializzata, inizierà l'apposita rubrica, sulle colonne di questa rivista, dedicata esclusivamente agli utilizzatori del package in questione.

Come procurarsi MSDOS 2

Oltre che in edicola, al prezzo di L.25000, i lettori potranno procurarsi il package direttamente dal nostro servizio arretrati inviando per posta la cifra richiesta (+ L.3000 per la spedizione semplice oppure + L.6000 per la spedizione raccomandata) mediante assegno bancario non trasferibile intestato semplicemente a: Systems Editoriale - Milano.

Coloro che, in possesso della precedente versione, allegheranno la prova di acquisto presente nella vecchia confezione, dovranno inviare, invece, una cifra inferiore, secondo quanto specificato nel talloncino stesso.

Stock
KLEENTECH
And clean up!
KLEENTECH
• Unique antistatic formulation that helps minimize dirt buildup
• "All over" screen & keyboard cleaner
• Won't streak screens
• Pleasant deodorising fragrance
• Pack can be safely adhered to the side of the VDU or operator's work station
KLEENTECH
KLEENTECH
15 PRE MOISTENED CLEANING TOWELS
15 PRE MOISTENED CLEANING TOWELS
• Non-woven towel won't fluff
• Residue free
• Won't harm plastic surfaces
• Can also be used for videos, TV sets, hi-fi equipment, type-writers, photostat machines, telephones
KLEENTECH
Soft care per il tuo hardware
Importatore esclusivo
I.B.C. Viale Corsica n. 72 Milano - Tel. 02/7387398 - 744698

Anche l'occhio vuole la sua parte

Qualche routine per migliorare la visualizzazione di messaggi sui vostri schermi

A cura di Alessandro de Simone

18800 Titolatrice
(Qualsiasi Commodore)

Avete mai avuto modo di osservare, in un film, la disposizione dei nomi dei protagonisti, soprattutto nei titoli di coda?

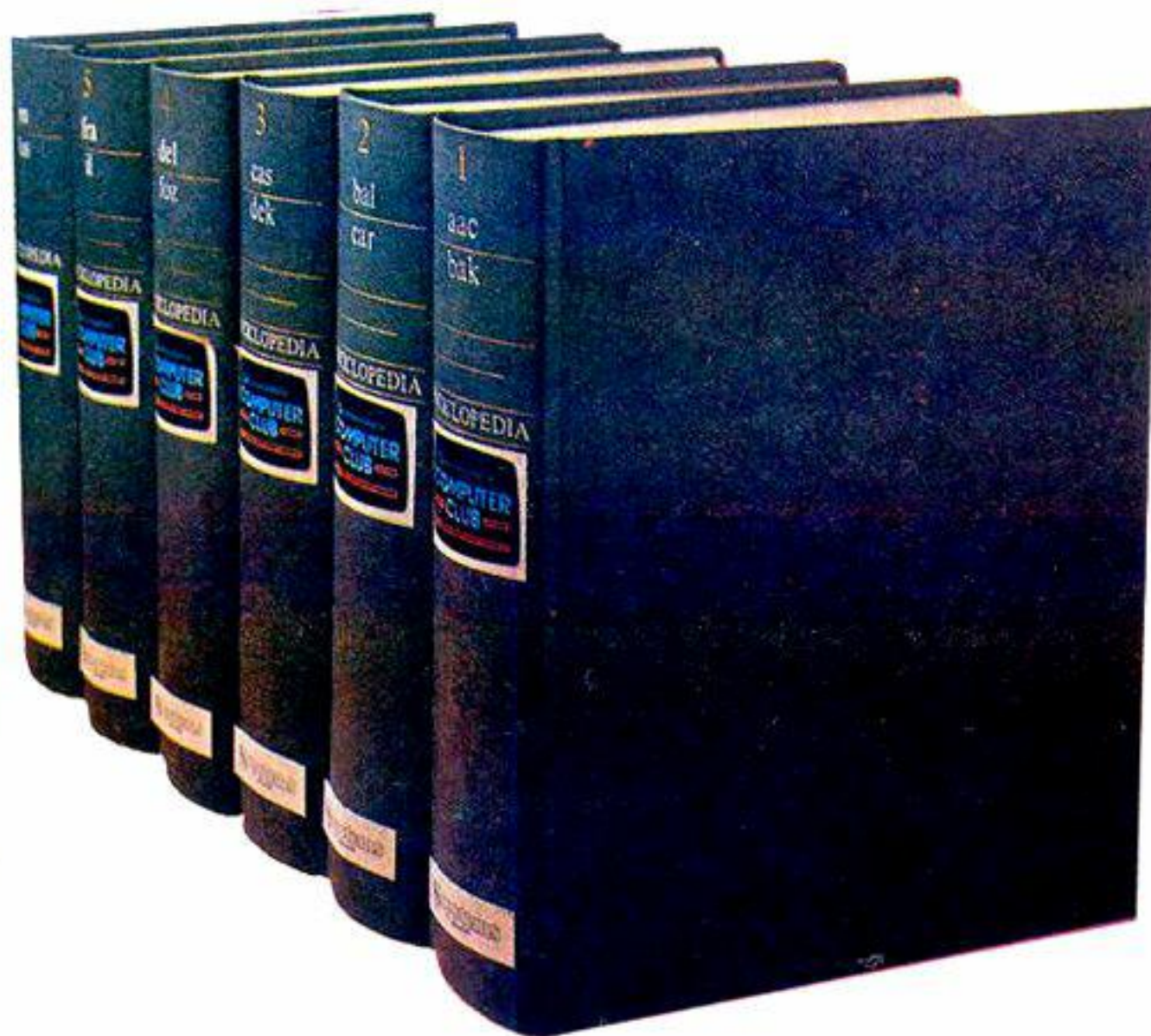
Molto spesso i nomi ed i cognomi degli attori sono disposti avendo come riferimento una colonna centrale: da questa "partono" i cognomi ed a questa "terminano" i nomi stessi.

Non è una configurazione a bandiera (che ha, come riferimento, il margine sinistro o destro) nè si può definire "centrata" dal momento che il nome ed il cognome dovrebbero distare un egual numero di spazi dai bordi estremi.

E', invece, una configurazione di testo che possiamo definire a "doppia bandiera": destra per i nomi e sinistra per i cognomi rispetto, ovviamente, ad una certa colonna presa come riferimento.

La routine che proponiamo si occupa proprio di assolvere questo compito: verifica che all'interno della stringa X1$ (lunga al massimo 39 caratteri) sia effettivamente presente un carattere di spazio e, quindi, spezza in due parti la stringa stessa; in seguito, dopo aver aggiunto un congruo numero di spazi alla sinistra del primo troncone (X8$) ed alla destra del secondo (X9$), ritorna al programma principale.

E' sufficiente stampare le due stringhe di seguito per vederle visualizzate incolonnate avendo come riferimento il numero X1 di colonna impostata prima di chiamare la routine.

Il demo, incorporato nel programma pubblicato, visualizza più volte i nomi di sei attori memorizzati per mezzo di Data. Si noti come la routine sia versatile e consenta, con minime modifiche, di ottenere anche incolonnamenti in diagonale.

```
   90 REM DEMO INCOLONNAMENTO NOM
      I
   91 :
  100 FOR X1=0 TO 35 STEP 7:
  110 FOR I=1 TO 6:READ X1$
  120 GOSUB 18800:PRINTX0$X8$X9$
  130 NEXTI:PRINT:RESTORE :NEXTX1
  131 :
  132 FOR X1=0 TO 35 STEP 7:
  133 FOR I=1 TO 6:READ X1$:X1=X1
      +1
  134 GOSUB 18800:PRINTX0$X8$X9$
  135 NEXTI:PRINT:RESTORE :NEXTX1
  136 :
  137 FOR X1=10 TO 40 STEP 20:
  138 FOR I=1 TO 6:READ X1$:X1=X1
      -1
  139 GOSUB 18800:PRINTX0$X8$X9$
  140 NEXTI:PRINT:RESTORE :NEXTX1
  150 :
  200 DATA  MARILYN MONROE
  210 DATA  MALCOM MC DOWELL
  220 DATA  GIANFRANCO D'ANGELO
  230 DATA  ELEONORA GIORGI
  240 DATA  PAOLO VILLAGGIO
  250 DATA  FRANCESCO SALVI
  260 :
 9999 END
18800 X8$="":X9$="":X0$="":IF X1<
      0 OR X1>39 THEN X0$="ERR":R
      ETURN
18805 FOR X9=1 TO 39:X9$=X9$+CHR$
      (32):NEXT:X1$=LEFT$(X1$+X9$
      ,39)
```

```
18810 X9=0:FOR X8=1 TO 39:IF MID$
      (X1$,X8,1)=CHR$(32) THEN X9
      =X8:X8=39
18820 X8$=X8$+MID$(X1$,X8,1):NEXT
18830 X8$=RIGHT$(X9$+X8$,X1+1)
18840 X9$=LEFT$(MID$(X1$,X9,39-X9
      ),39-LEN(X8$))
18860 RETURN
18885 REM X1=DISTANZA DELLO SPAZI
      O DAL BORDO
18886 REM X1$=(INPUT) NOME CONTEN
      ENTE UNO SPAZIO
18887 REM X8$= (OUTPUT) NOME SINI
      STRO
18888 REM X9$= (OUTPUT) NOME DEST
      RO
18899 REM INCOLONNAMENTO NOMI CON
       SPAZIO
```

```
18900 X0$="":IF X1>25 OR X1<0 THE
      N X0$="ERR":RETURN
18905 X3$="":X3=0:X4=LEN(X1$)
18910 GET X2$:IF X2$=CHR$(13) THE
      N 18940
18920 IF X2$<>"[DOWN]" AND X2$<>"
      [UP]" AND X2$<>"[RIGHT]" AN
      D X2$<>"[LEFT]" THEN 18910
18930 X3$=X3$+X2$:X3=X3+1:IF X3<X
      4 THEN 18910
18940 PRINTCHR$(147):FOR X3=1 TO
      X1:PRINT:NEXT
18950 FOR X1=1 TO X4:FOR X3=1 TO
      X2:NEXTX3
18960 PRINTMID$(X3$,X1,1)MID$(X1$
      ,X1,1);:NEXTX1:RETURN
18999 REM VSUALIZZA SALTELLANDO
```

18900 Messaggi mobili
(Qualsiasi Commodore)

Con questa divertente routine sarà possibile visualizzare, carattere per carattere, un qualsiasi messaggio ad una velocità dipendente dal valore X2 ed a partire dal rigo video indicato da X1.

Naturalmente la visualizzazione avverrà in un modo piuttosto particolare ed, esattamente, in accordo al contenuto della stringa X3$ che può ospitare, grazie ad un particolare Input controllato, solo i comandi relativi ai quattro movimenti del cursore.

Il Demo inserito nella stessa routine chiarirà l'uso che si può fare del sottoprogramma meglio di un fiume di parole: quando appare il messaggio "Premi i tasti cursore..." divertitevi a premere, nella successione che preferite, la combinazione dei tasti cursore, con e senza shift.

Non appena avrete battuto il tasto Return, oppure avrete raggiunto la lunghezza del nome contenuto in X1$, vedrete che il messaggio verrà visualizzato, carattere per carattere, secondo la successione dei tasti premuti.

```
100 X1$="COMMODORE COMPUTER CLU
    B":REM  NOME
110 X1=10:REM  N.RIGO INIZIO
115 X2=100:REM  VELOCITA' VISUA
    LIZZAZIONE
120 PRINT"PREMI I TASTI CURSORE
     PER"LEN(X1$)"VOLTE"
130 GOSUB 18900
9998 :
9999 END
```

19000 Attesa ipnotica
(Qualsiasi Commodore)

Questa routine, scritta per C/64, può essere facilmente adattata al C/16 ed al Plus/4 modificando le righe 19000 e 19010 come indicato, rispettivamente, nelle Rem di righe 19001 e 19011.

Non fa altro che visualizzare casualmente un gruppo di caratteri, tutti eguali, disposti però con una certa regolarità e, in certi casi, con piacevole simmetria.

A seconda del valore attribuito alla variabile X1, la visualizzazione continua all'infinito (X1=0) finchè non si preme un tasto; altrimenti continua per il numero di volte stabilito con X1 stessa, a meno che non si prema un tasto qualsiasi prima che giunga al termine del conteggio.

Può essere utile per attrarre l'attenzione dell'utilizzatore, oppure per... ipnotizzarlo!

```
  90 REM DEMO ATTESA IPNOTICA
  91 :
 100 X1=0:REM CICLO INFINITO
 105 PRINTCHR$(147)"PER INTERROM
     PERE L'EFFETTO CHE ORA VEDR
     AI, PREMI UN TASTO"
 106 GET A$:IF A$="" THEN 106
 110 GOSUB 19000
 120 PRINTCHR$(147):INPUT "QUANT
     E VOLTE";X1
 130 GOSUB 19000
9998 :
9999 END
```

```
19000 PRINTCHR$(147):X4=1023:X2=I
      NT(RND(0)*50):IF X2<20 THEN
       19000
19001 REM : C/16: PRINTCHR$(147):
      X4=3071:X2=INT(RND(0)*50):I
      FX2<20THEN19000
19005 X5=RND(0)*255:IF X5=32 OR X
      5=160 THEN X5=33
19010 X4=X4+X2:IF X4<2023 THEN PO
      KE X4,X5:GOTO 19010
19011 REM : C/16: X4=X4+X2:IFX4<4
      070THENPOKEX4,X5:GOTO19010
19015 GET X1$:IF X1$<>"" THEN RET
      URN
19020 IF X1>0 THEN X3=X3+1:IF X3<
      X1 THEN 19000
19040 IF X1=0 THEN 19000
19050 RETURN
19099 REM ATTESA IPNOTICA
```

19100 Somma di ore
(Qualsiasi Commodore)

Vi è mai capitato di dover sommare, tra loro, due orari? Il problema non è molto semplice dal momento che la suddivisione del tempo non è in notazione decimale, ma sessagesimale (per ore e minuti), per non parlare del giorno, suddiviso in 24 ore.

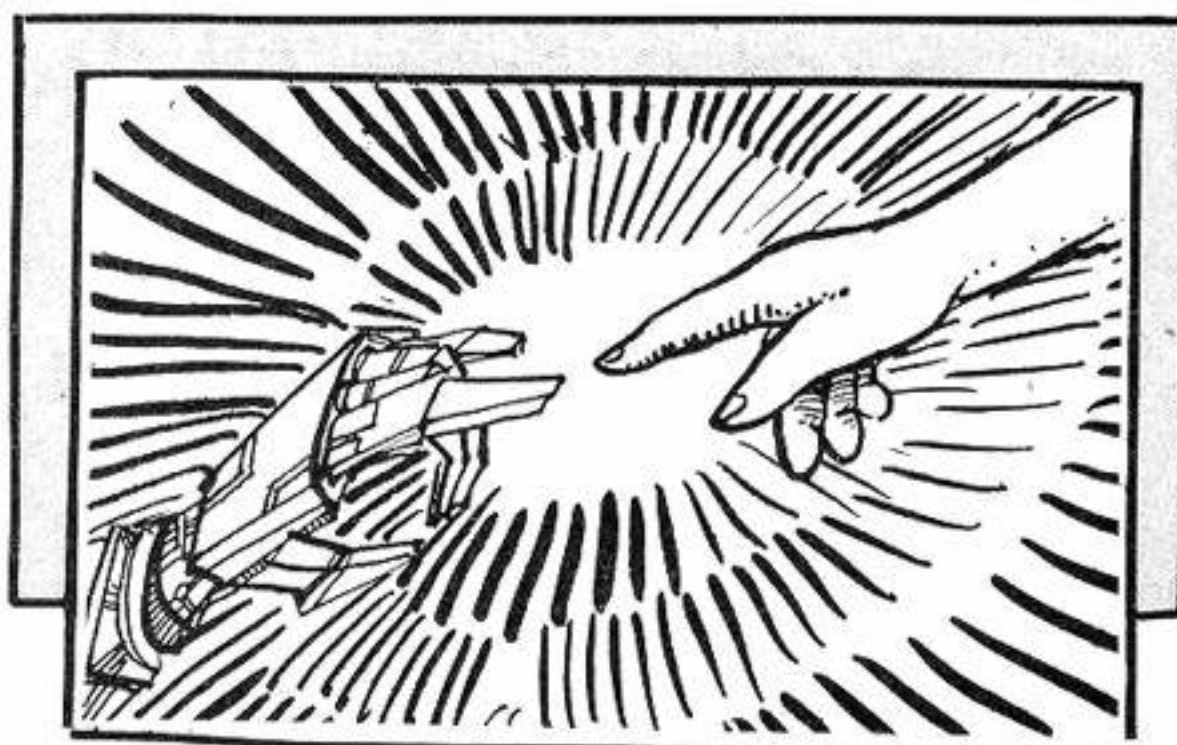
Con la routine di queste pagine sarà possibile assegnare alla variabile X1$ il primo dei due orari ed alla variabile X2$ il secondo; il formato è quello ben noto della variabile TI$, formata da sei caratteri che, a coppia di due, rappresentano le ore, i minuti ed i secondi.

La routine contiene, al suo interno, alcuni controlli che consentono di individuare errori di formato ed incongruenza di orari come, ad esempio: 256399.

Il demo effettua, e visualizza, la somma tra 124535 e 115456; subito dopo chiederà altre somme da effettuare.

Si noti che se la somma dei due orari (assegnata alla variabile X3$) supera le 24 ore, la variabile X0 sarà posta ad uno; in caso contrario varrà zero.

Si tenga presente, inoltre, che, al ritorno dalla routine, i primi tre elementi del vettore X$(I) contengono, rispettivamente, le ore, i minuti ed i secondi elaborati.

```
 90 REM SOMMA DI ORE
 91 :
100 X1$="124535"
110 X2$="115456"
115 GOSUB 19100:PRINTX1$"+"
120 PRINTX2$"=":PRINTX3$:PRINT"
    LA VARIABILE X0 VALE"X0
130 INPUT "ORARIO 1 (ES:123425)
    ";X1$
140 INPUT "ORARIO 2 (ES:123425)
    ";X2$
150 GOSUB 19100
160 IF X0$="" THEN PRINT"SOMMA
    ORE:":PRINTX1$"+":PRINTX2$;
    "=":PRINTX3$
170 IF X0>0 THEN PRINT"OLTRE AD
     UN INTERO GIORNO"
180 GOTO 130
9998 :
9999 END
19100 X0$="":X0=0:IF LEN(X1$)<>6
      OR LEN(X2$)<>6 THEN X0$="ER
      R":RETURN
19105 FOR X1=1 TO 6 STEP 2
19110 X1(X1)=VAL(MID$(X1$,X1,2)):
      X2(X1)=VAL(MID$(X2$,X1,2)):
      NEXT
19120 IF X1(1)>23 OR X2(1)>23 OR
      X1(3)>59 OR X2(3)>59 THEN X
      0$="ERR":RETURN
19140 IF X1(5)>59 OR X2(5)>59 THE
      N X0$="ERR":RETURN
19150 X(3)=X1(5)+X2(5):X(2)=X1(3)
      +X2(3):X(1)=X1(1)+X2(1)
19160 IF X(3)>59 THEN X(3)=X(3)-6
      0:X(2)=X(2)+1
19170 IF X(2)>59 THEN X(2)=X(2)-6
      0:X(1)=X(1)+1
19179 IF X(1)>23 THEN X(1)=X(1)-2
      4:X0=1
19180 FOR X9=1 TO 3:X3$=STR$(X(X9
      ))
19185 X$(X9)=RIGHT$("00"+RIGHT$(X
      3$,LEN(X3$)-1),2):NEXT
19190 X3$="":FOR X1=1 TO 3:X3$=X3
      $+X$(X1):NEXT:RETURN
19199 REM  SOMMA DI DUE ORARI (X1
      $+X2$=X3$)
```

Elenco delle ultime routine pubblicate

63902 rem 18600 incorniciatore (45)
63903 rem 18500 string writer (45)
63904 rem 18400 password (45)
63905 rem 18300 string play (45)
63906 rem 18200 fusi orari (44)
63907 rem 18100 paroliere (44)
63908 rem 18000 random programmab. (44)
63909 rem 17900 gestione menu (44)
63910 rem 17800 legge file (43)
63911 rem 17700 memo-messaggi (43)
63912 rem 17600 amiga menu (43)
63913 rem 17500 mini text-editor (42)
63914 rem 17400 h/copy low-res (42)
63915 rem 17300 grand. analogiche (41)
63916 rem 17200 interp A$ (41)
63917 rem 17100 equivalenze (40)
63918 rem 17000 percentuali (40)
63919 rem 16900 deek & doke (39)
63920 rem 16800 sprite scanner (39)
63921 rem 16700 movimento sprite (39)
63922 rem 16600 accensione sprite (39)
63923 rem 16500 drum per c/64 (38)
63924 rem 16400 draw low/res (38)
63925 rem 16300 print v/cont (38)
63926 rem 16200 plot low-res (37)
63927 rem 16100 integrali (37)
63928 rem 16000 equaz. mista (37)
63929 rem 15900 equaz. terzo gr. (37)
63930 rem 15800 derivata di funz. (37)
63931 rem 15700 scritte rotanti (37)
63932 rem 15600 convers. coordin. (36)
63933 rem 15500 logar. base quals. (36)
63934 rem 15400 conversione basi (36)
63935 rem 15300 semplif. frazioni (36)
63936 rem 15200 divis. con N decim.(36)
63937 rem 50100 directory (35)
63938 rem 15100 lampeggio righe (35)
63939 rem 15000 frammenta schermo (35)
63940 rem 14900 delete window (35)
63941 rem 14800 cambia stringhe (34)
63942 rem 14700 slitta stringhe (34)
63943 rem 14600 ruota stringhe (34)
63944 rem 10500 input programmab. (34)
63945 rem 14500 scroll solo testo (33)
63946 rem 14400 sprite multiuso (33)
63947 rem 14300 zoom esadecimale (33)
63948 rem 14200 video orologio (33)
63949 rem 11100 funzioni inverse (32)
63950 rem 13200 centra messaggi (32)
63951 rem 14100 finestre di testo (32)
63952 rem 14000 gestione nome disk (32)
63953 rem 13900 load/save pg.video (31)
63954 rem 13800 scritte in ebcm (31)
63955 rem 13700 bit image mps/803 (31)
63956 rem 13600 or esclusivo (31)
63957 rem 13500 comandi extra prg (31)
63958 rem 13400 linee low-res. (31)
63959 rem 13300 elabora stringhe (31)
63960 rem 13200 centratura frase (32)
63961 rem 13100 menu con joy (30)
63962 rem 13000 menu con cursore (30)
63963 rem 12900 frase lampeggiante (29)
63964 rem 12800 bordo technicolor (29)
63965 rem 12700 fill memoria ram (29)
63966 rem 12600 text copy mps 803 (29)
63967 rem 12500 colore pag.testo (29)
63968 rem 12400 print using (31)
63968 rem 12400 print using (29)
63969 rem 12300 m.c.d. E M.C.M. (29)
63970 rem 50500 visualizza file (28)
63971 rem 50400 read file relativi (28)
63972 rem 50300 write file relat. (28)
63973 rem 50200 crea file relativi (28)
63974 rem 50000 legge blocks free (28)
63975 rem 12200 numeri congrui (28)
63976 rem 12100 protezione S/W (28)
63977 rem 12000 koala (27)
63978 rem 11900 change pag. video (27)
63979 rem 11800 salva ram (27)
63980 rem 11700 calcolatrice (27)
63981 rem 11600 scomposiz.sillabe (27)
63982 rem 11500 car.hi-res (27)
63983 rem 11400 istogrammi (27)
63984 rem 50100 esame directory (26)
63985 rem 11300 funz.inv.iperbol. (26)
63986 rem 11200 funz.inv.trigonom. (26)
63987 rem 11100 funzioni inverse (26)

Come realizzare l'enciclopedia e utilizzarla nei propri listati.

Ai lettori che hanno acquistato per la prima volta questo numero di Commodore Computer Club, illustriamo qui di seguito, in breve, i vantaggi derivanti dalla raccolta proposta. Questa, a pensarci bene, è la versione "superiore" della rubrica "1 RIGA" e potrebbe anche denominarsi... "Una schermata"!

Oltre che utili per costituire un'enciclopedia, i brevissimi sottoprogrammi pubblicati su ogni numero, sono anche validissimi strumenti di studio per coloro che desiderano approfondire le proprie conoscenze del Basic, esaminando, senza fatica, particolari routine o insolite tecniche di programmazione.

- Dato che può esser "chiamata" più di una volta nel corso di un programma, nessuna routine contiene istruzioni del tipo DATA oppure DIM, allo scopo di non creare confusione col listato principale.
- Nessuna routine può far riferimento ad altre routine dell'enciclopedia.
- Nessuna routine può contenere variabili "banali" (A, A$, eccetera), ma solo variabili poco usate (X1$, X8, Y0%, eccetera).
- Ogni routine deve apparire, **per intero,** sullo schermo del computer e consentire, proprio per questo motivo, di essere esaminata comodamente.
- Ogni routine deve esser numerata secondo uno standard che ha la particolarità di esser ricordato facilmente:

Righe	Contenuto
XXY00	Prima riga del sottoprogramma
XXY89	Ultima riga utile del sottoprogramma
XXY90 REM	Prima riga di spiegazioni
XXY99 REM	Nome della subroutine

in cui XX sono due valori variabili da 10 a 63; Y è un carattere numerico compreso tra 0 e 9.

Qualsiasi subroutine, in altre parole, inizia con un numero, di cinque caratteri, che termina **sempre** con "00". La stessa subroutine, d'altra parte, ha l'ultima riga numerata con "99". Digitando, ad esempio: LIST 10800-10899
si avrà la certezza di veder apparire sullo schermo, **per intero,** la routine il cui nome si trova nella riga 10899.

Prima di accedere alla routine, è necessario assegnare, alle variabili indicate con REM da riga XXY89 a XXY98, particolari valori per il suo corretto funzionamento. Al "ritorno" una o più variabili conterranno il risultato dell'elaborazione.

In questo modo, per esser più chiari, è possibile simulare alcuni comandi di versioni Basic avanzate oppure, addirittura, creare nuove e inedite istruzioni. Ad esempio, il comando: SOUND 1,800,500
che, nel C-16, riproduce un suono di tonalità 800 tramite la voce 1 per la durata 500, potrebbe venir riprodotta, in un'ipotetica subroutine per il Commodore 64, con: X1=1:X2=800:X3=500:GOSUB12400
nell'ipotesi, ovviamente, che la routine in oggetto sia allocata da riga 12400 a 12499.

I listati pubblicati "girano" su ogni computer, salvo dove indicato diversamente.

E' ovvio che nel caso del Vic-20, (che, come è noto, ha uno schermo di soli 506 caratteri), le subroutine "universali" funzionano correttamente, ma non possono apparire per intero in una sola schermata.

Per quanto riguarda la digitazione, si tenga presente che sulla rivista, per motivi di chiarezza, i comandi e le istruzioni Basic sono separati tra loro da spazi bianchi. Nel digitare le linee di programma, pertanto, è opportuno ignorarli altrimenti si rischia di non restare in una sola schermata. Se, per esempio, leggete:

```
12100 X1=34:  X2  = SQR(X3) + LOG(X1)
```

digitate nel modo seguente:

```
12100 X1=34:X2=SQR(X3)+LOG(X1)
```

senza, cioè, alcun carattere di separazione tra comandi ed istruzioni.

Collaborazione dei lettori

La collaborazione dei lettori è gradita, purchè si provveda a inviare **almeno** tre sottoprogrammi per volta, su nastro, disco oppure output di stampante. I listati di routine che non rispettano lo standard adottato non potranno esser presi in considerazione.

Tutti i lavori pubblicati verranno compensati con prodotti della Systems Editoriale (cassette di programmi, libri, abbonamenti, copie arretrate, eccetera).

Tanti giochi e due utility

Il numero 16 di Software club contiene tanti giochi divertenti; ma anche una Ram-Disk per C/64-128

Vediamo in dettaglio i titoli dell'ultimo numero di "Software Club" per la parte dedicata ai possessori di C/64-128.

Cover

Come tutti i numeri di Software Club, il primo programma è una simpatica presentazione del contenuto della cassetta.

Over the top

Il tema centrale del film "Over the top" riguarda i campionati nazionali di "Arm-wrestling", braccio di ferro.

In questo videogame sarai tu che, joystick alla mano, dovrai sfidare i campioni di questo fantastico sport.

Prima dell'inizio del match l'indicatore nella parte alta dello schermo è fermo nella posizione centrale, che equivale alla condizione di parità durante il gioco.

Premendo FIRE inizia il match.

Muovendo velocemente lo stick del joy alternativamente a destra e a sinistra si contrasta lo sforzo dell'avversario, che tende a muovere l'indicatore verso destra.

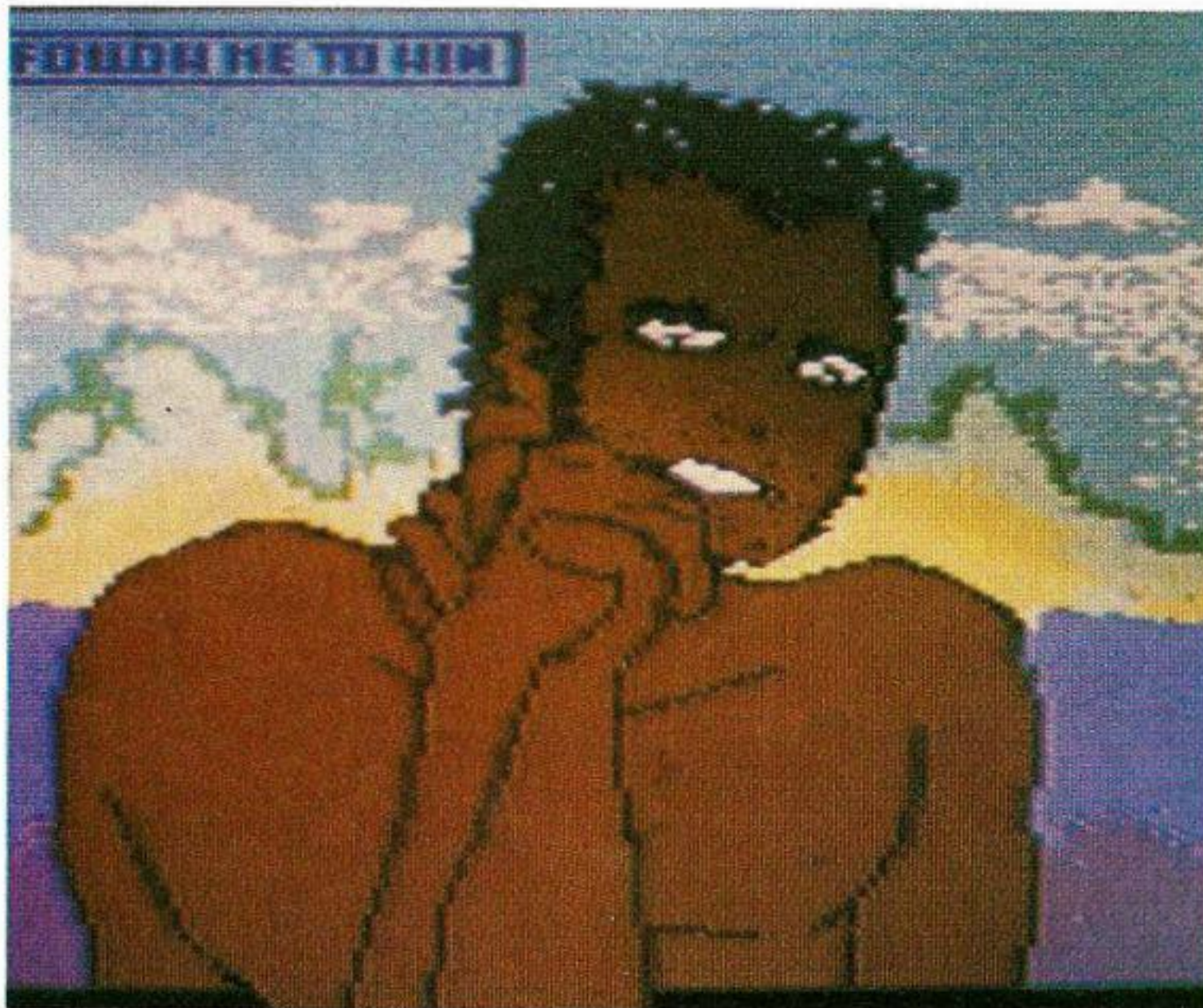
Per il giocatore il match si divide in due fasi possibili:

1/ Fire non premuto - In questo caso è più facile contrastare l'attacco dell'avversario, ma l'indicatore non può essere spinto a sinistra oltre il punto centrale. Questa è la fase di DIFESA.

2/ Fire premuto - In questo modo l'attacco dell'avversario è più potente, ma il giocatore può vincere muovendo velocemente lo stick, tanto da spingere l'indicatore completamente a sinistra. Questa è la fase di ATTACCO.

Ad ogni vittoria del giocatore l'avversario diventa sempre più forte, sia in fase di attacco che in fase di difesa.

Ogni passaggio di livello è identificato da un particolare commento che compare in alto a sinistra nella schermata intermedia.

I livelli sono più di dieci, ma oltre il nono la vittoria risulta praticamente impossibile per i limiti meccanici del joystick.

Croll ship

Lo scopo del gioco è quello di colpire le 30 "E" rotanti prima di colpire la stella contrassegnata con "FINAL", al centro della grotta.

Il nemico, che insegue l'astronave, va preferibilmente evitato piuttosto che colpito, poichè quando viene colpito il carico dell'astronave diminuisce.

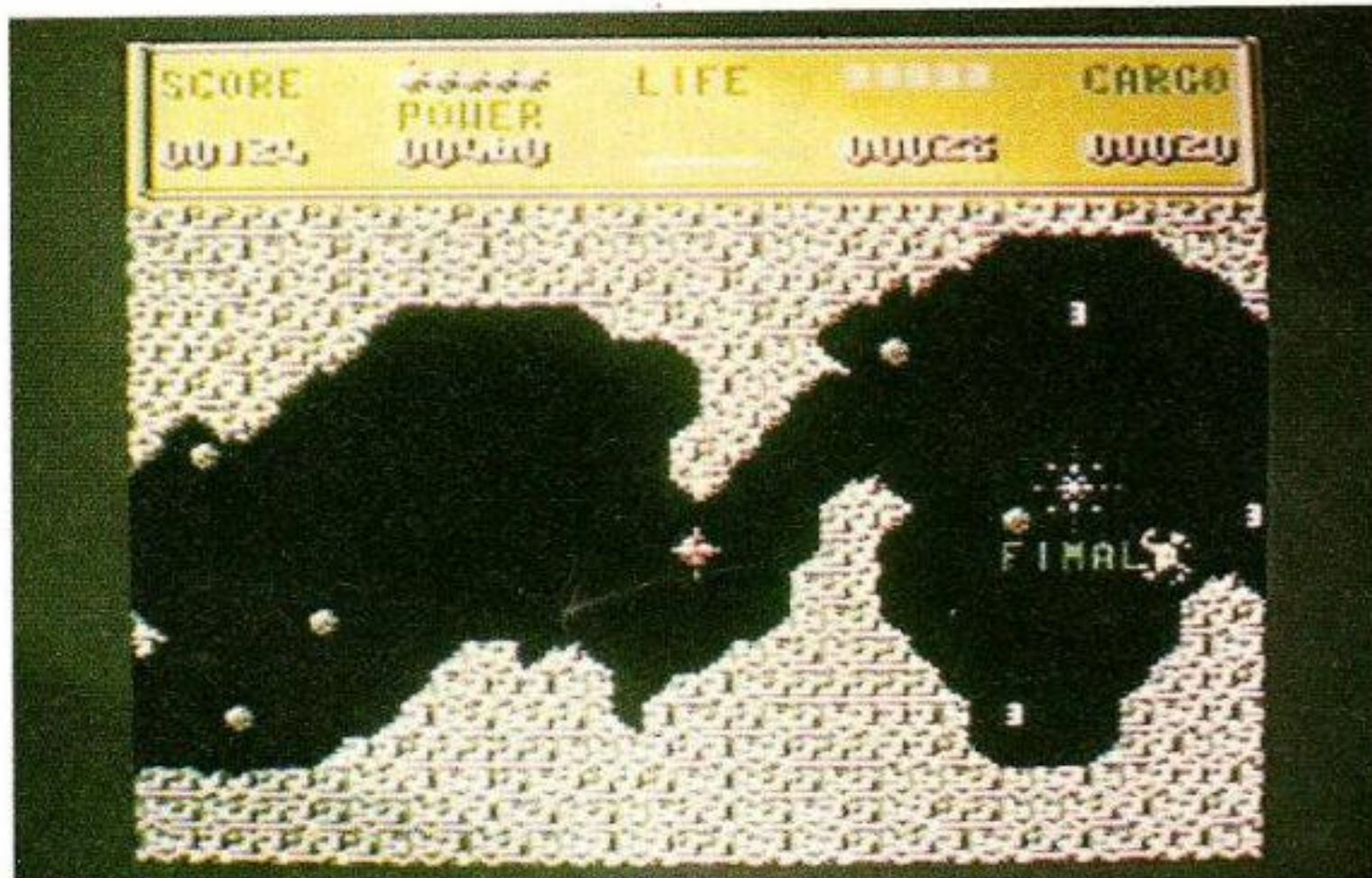
Il carico è costituito dal numero di unità di energia colpite (le "E" rotanti) e viene aggiunto al punteggio ogni volta che l'astronave torna alla base ("Energy Home").

La potenza dei nemici, che influenza la frequenza con cui appaiono le pietre-ostacolo all'interno della grotta, aumenta a mano a mano che si raccolgono energie.

Ad ogni ritorno alla base la potenza dei nemici viene decrementata.

I livelli sono tre e differiscono per la distribuzione dei bersagli e la forma della grotta che rendono più ardua la raccolta delle unità di energia.

Il passaggio di livello avviene quando, dopo aver raccolto tutte le unità di energia, si colpisce la stella posizionata al centro della grotta.

La parte grafica dei due giochi, molto ben curata, non ha nulla da invidiare ad altri videogame consimili.

Deltaplane

Il deltaplano, uno dei velivoli più affascinanti, ispira questo gioco divertente.

Con il Joy in porta 2 controlli il deltaplano dal quale lancerai bombe con lo scopo di colpire tutte le bandierine presenti nel paesaggio.

Attenzione! Sarà possibile colpire le bandierine solo avendo cura che la bomba esploda nel momento della collisione con la bandierina stessa.

Se proprio non riesci a colpire le bandierine, beh, almeno goditi il paesaggio...

Blob

Ed ecco un altro gioco molto ben curato che poco ha da invidiare alla concorrenza d'oltre oceano.

Una simpatica ape golosa di... ciliege si muove in un labirinto a caccia dei frutti preferiti.

Ma in agguato ci sono i guardiani che cercheranno in tutti i modi di fermarla.

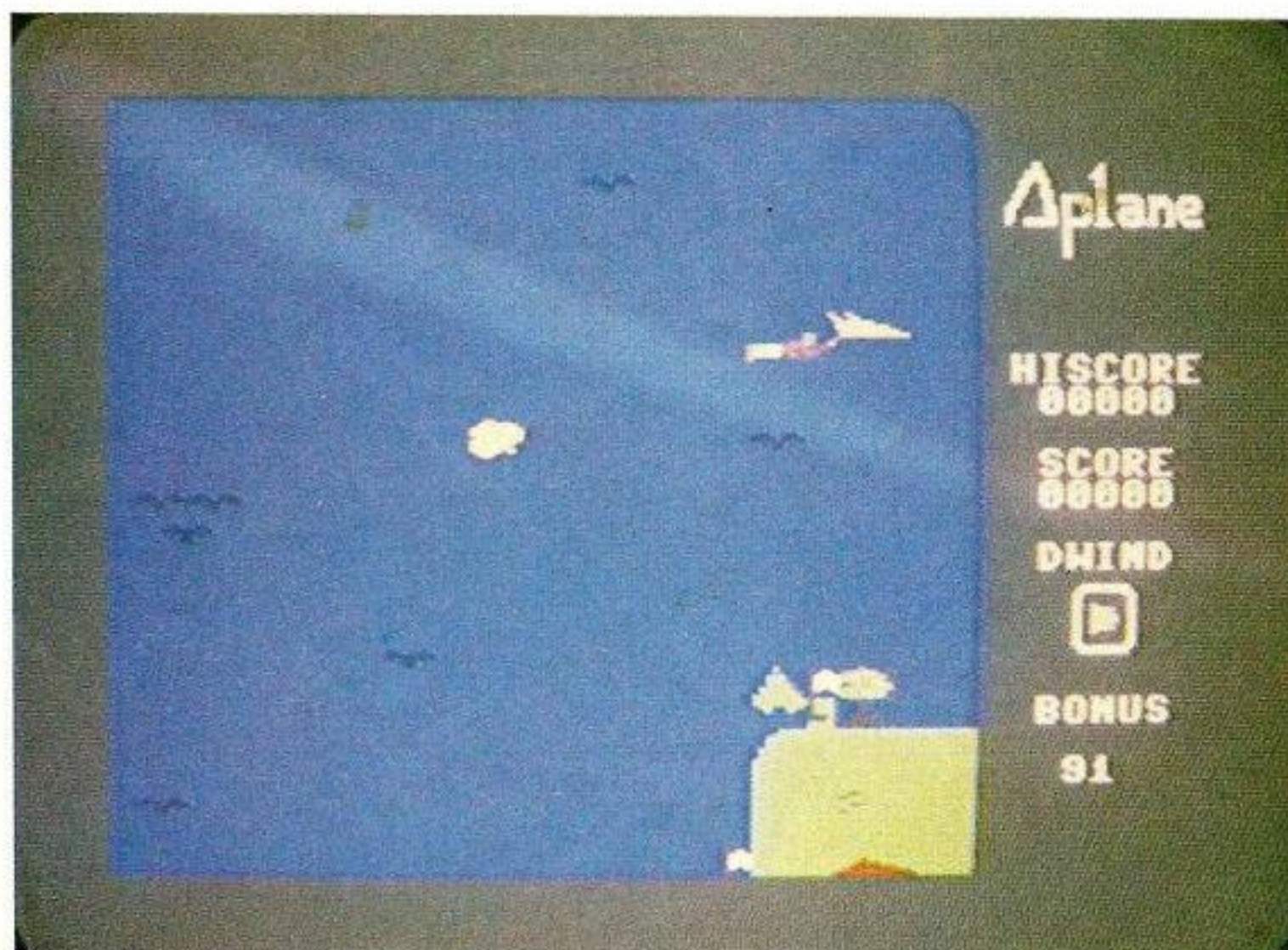
Rispondendo "Y" alla domanda "Do you want to make your selection?" sarà possibile selezionare varie opzioni:

Speed (1-4): velocità del gioco
N. of enemies (2-4): numero dei nemici
N. of cherries (5-40): numero delle ciliege

Decisione e destrezza occorrono per questo gioco che contiene tantissimi schermi differenti.

Ram-Disk by 4WD-Soft

Ed eccoci arrivati al "clou" della cassetta; la Ram-Disk è il risultato di un lungo ed attento lavoro di programmazione che ha premiato gli autori con una notevole soddisfazione.

Tutti coloro che utilizzano computer evoluti, come IBM, Atari, Amiga e così via, hanno certamente utilizzato una Ram-Disk.

La quasi totalità di utilizzatori di C/64-128, invece ignora quasi del tutto l'esistenza di questo potente strumento di lavoro.

Che cos'è una Ram-Disk?

Si tratta di un'area di memoria RAM che viene protetta da scrittura e che viene usata a tutti gli effetti come se fosse una periferica (virtualmente un disk-drive).

I vantaggi di questa partizione di memoria sono numerosi: in primo luogo, durante il lavoro di programmazione, non sarà necessario salvare continuamente su disco o nastro tutte le varie versioni del programma (con notevole perdita di tempo); sarà possibile salvarle temporaneamente in memoria per poi salvare a fine lavoro un'unica versione su disco o nastro.

In secondo luogo, il programma è protetto da qualsiasi cancellazione accidentale, tranne il caricamento di un programma troppo lungo (superiore ai 110 blocchi).

Sia Run/stop e Restore, che il Reset, non alterano il funzionamento del programma per cui il comando "R" (freccia a sinistra + R) è sempre presente e permette di passare dal Basic al modo Ram-Disk.

In terzo luogo funziona da "espansione": infatti, la memoria per il Basic è limitata a $7FE0 (circa 30 KB) e lo spazio per la Ram-Disk è di circa 18 KB. In pratica con questa utility si hanno 30+18 KB = 48 KByte (non male!)...

La Ram-Disk viene gestita con nove comandi da visualizzare con il comando HELP:

• L "nome prg": carica da RAM il programma.
• S "nome prg": salva il programma nella zona RAM.
• V "nome prg": esegue la verifica del programma.
• D: elenca i programmi presenti indicandone anche l'occupazione in byte.
• F: formatta l'area di memoria cancellando gli eventuali programmi presenti.
• R "nome prg": rinomina il file specificato.
• E "nome prg": cancella il programma in questione.
• H: elenca i comandi.
• Q: ritorna al Basic.

Esempi d'uso

Supponiamo che abbiate registrato su disco, o nastro, un programma denominato "Pacchio" e che vogliate modificarlo.

Per prima cosa sarà opportuno caricare e lanciare il programma Ram-Disk; successivamente sarà possibile caricare il programma "Pacchio", da disco o nastro, quindi digitare il comando di raname (R) e, una volta in ambiente Ram-Disk, dare il comando S "Pacchio".

A questo punto il programma "Pacchio" è in memoria (sia nell'area Basic sia nell'area Ram-Disk).

Tramite il comando Q sarà possibile tornare al Basic e modificare il listato originale e, quando lo si ritiene opportuno, salvarlo sotto Ram-Disk anzichè su periferica.

In questo modo si risparmierà una notevole quantità di tempo.

Una volta terminata la fase di modifica o miglioramento del programma sarà possibile salvarlo definitivamente su supporto magnetico.

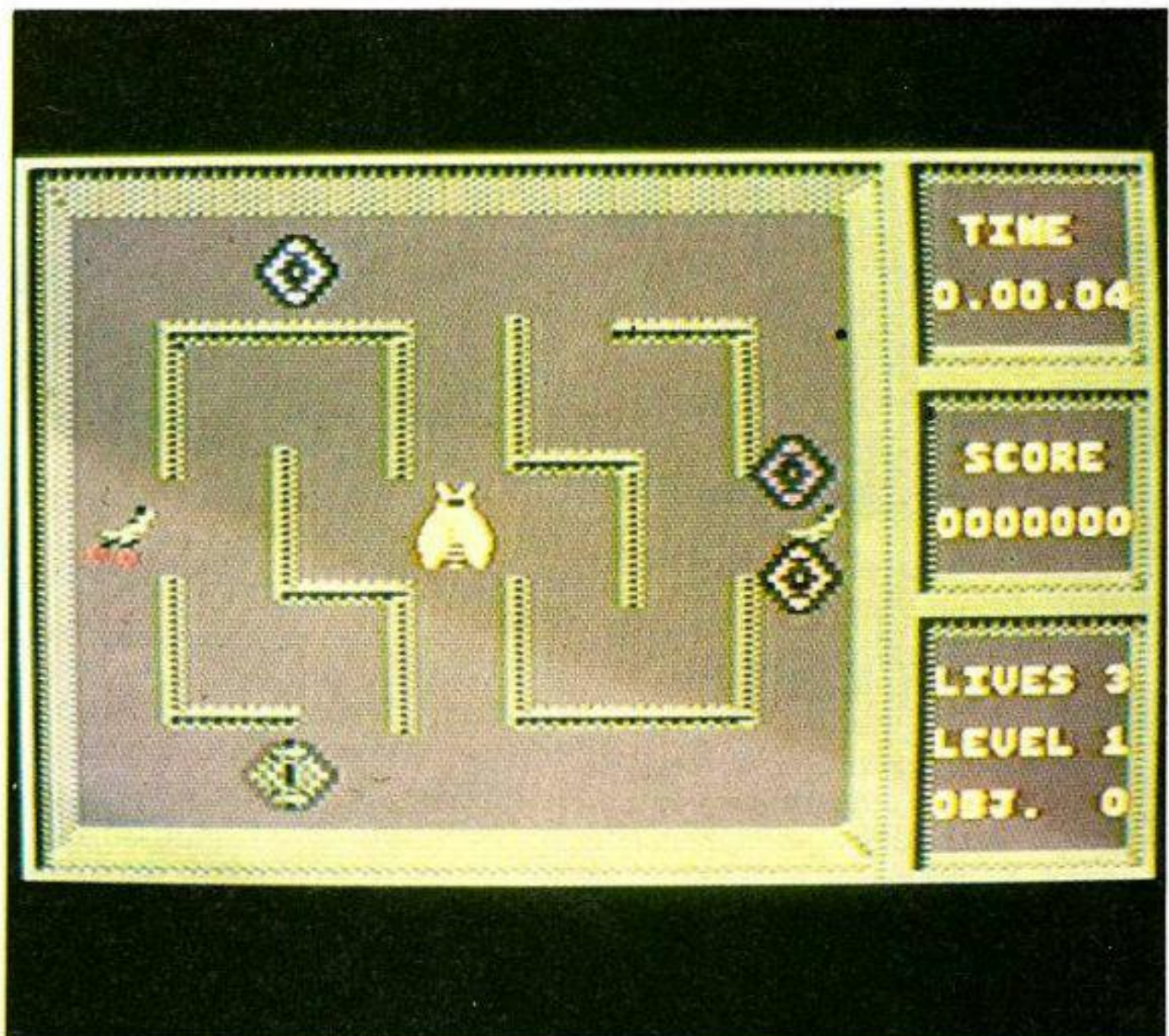
Come il lettore certamente saprà, il numero dei salvataggi (e caricamenti) di un programma di media lunghezza, prima della sua versione definitiva, è piuttosto elevato (noia a parte); tramite l'utility proposta sarà quindi possibile non solo evitare i "tempi morti" ma anche un'usura eccessiva della periferica.

Marker

La seconda utility di questo numero ha lo scopo di facilitare la stesura e il controllo di programmi molto lunghi.

Una volta attivata, verrà visualizzata accanto al cursore una striscia, lunga 40 caratteri, di colore bianco, interamente formata da sprite.

Lo scopo di questa striscia è quello di evidenziare la riga corrente in cui si trova il cursore.

Per attivare o disattivare l'utility è sufficiente impartire il comando SYS 49253

Software Club

Eccoci ora arrivati alla "zona riservata" al C-16/ Plus/4

My little pet

State esplorando la piramide di Tutankamon, quando scoprite un passaggio segreto che vi porta in un labirinto sotterraneo disseminato di rare gemme.

Una volta entrati, il passaggio si chiude e rimanete intrappolati: non esistono altre vie di uscita.

Nelle vicinanze del punto in cui siete entrati vedete, sul muro, un'antica iscrizione in rilievo che dice di raccogliere tutte le gemme per poter uscire dal labirinto, ma in caso di fallimento quella stanza sarà la vostra tomba.

Nell'esplorare il labirinto fate attenzione a non urtare le pareti per non perdere rapidamente le vostre energie.

Per muoversi, usare il joystick in porta 1.

Quickfish

Una piattaforma petrolifera mette in pericolo la vita tranquilla di un simpatico pesciolino.

Il vostro compito è quello di aiutarlo a mettere in salvo le sue uova prima che si verifichi un disastro ecologico causato dalla rottura delle tubazioni dell'impianto.

L'unica possibilità di salvezza per le uova è una cozza gigante che troverete superando una montagna sommersa; fate però attenzione a non rimanervi intrappolati.

Prestate attenzione nel superare la montagna: in cima ad essa passano alcune bisce marine voracissime!

Per prendere, e depositare, le uova premete il pulsante di fuoco del joystick, o la barra spaziatrice.

Per muoversi, invece, usare il joystick od i tasti cursore.

Shot

La vostra città sta subendo un raid aereo, e le forze nemiche si stanno paracadutando per occuparla.

Presi alla sprovvista, tutti i tentativi di difesa sono risultati inutili; l'ultima speranza è di evitare che i paracadutisti atterrino sulla città.

La vostra fama di tiratore scelto fa sì che vi venga affidato l'incarico di abbattere i numerosi nemici che si avvicinano velocemente alla meta.

Avrete a disposizione il modernissimo cannone-laser XL-412, che potrete manovrare con i seguenti tasti:

- Y: sposta verso l'alto
- B: sposta verso il basso
- G: sposta verso destra
- H: sposta verso sinistra
- Spazio: spara

Anche se i colpi a disposizione sono numerosi, è consigliabile non sprecarne perchè i nemici non danno tregua.

Cruncher

Una fantastica utility dedicata però ai soli possesori del disk drive.

Si tratta di un programma che per-

mette di compattare qualsiasi tipo di file (Programmi, testi, schermate e così via), in modo da risparmiare spazio sui dischetti.

Alla domanda "Source File Name", dovremo rispondere con il nome del file che intendiamo compattare e alla domanda "Object File Name" risponderemo con il nome che vorremo assegnare al file compattato.

Per ricaricare il file compattato è sufficiente caricarlo normalmente quindi dare RUN per dare inizio alla procedura di "scompattamento".

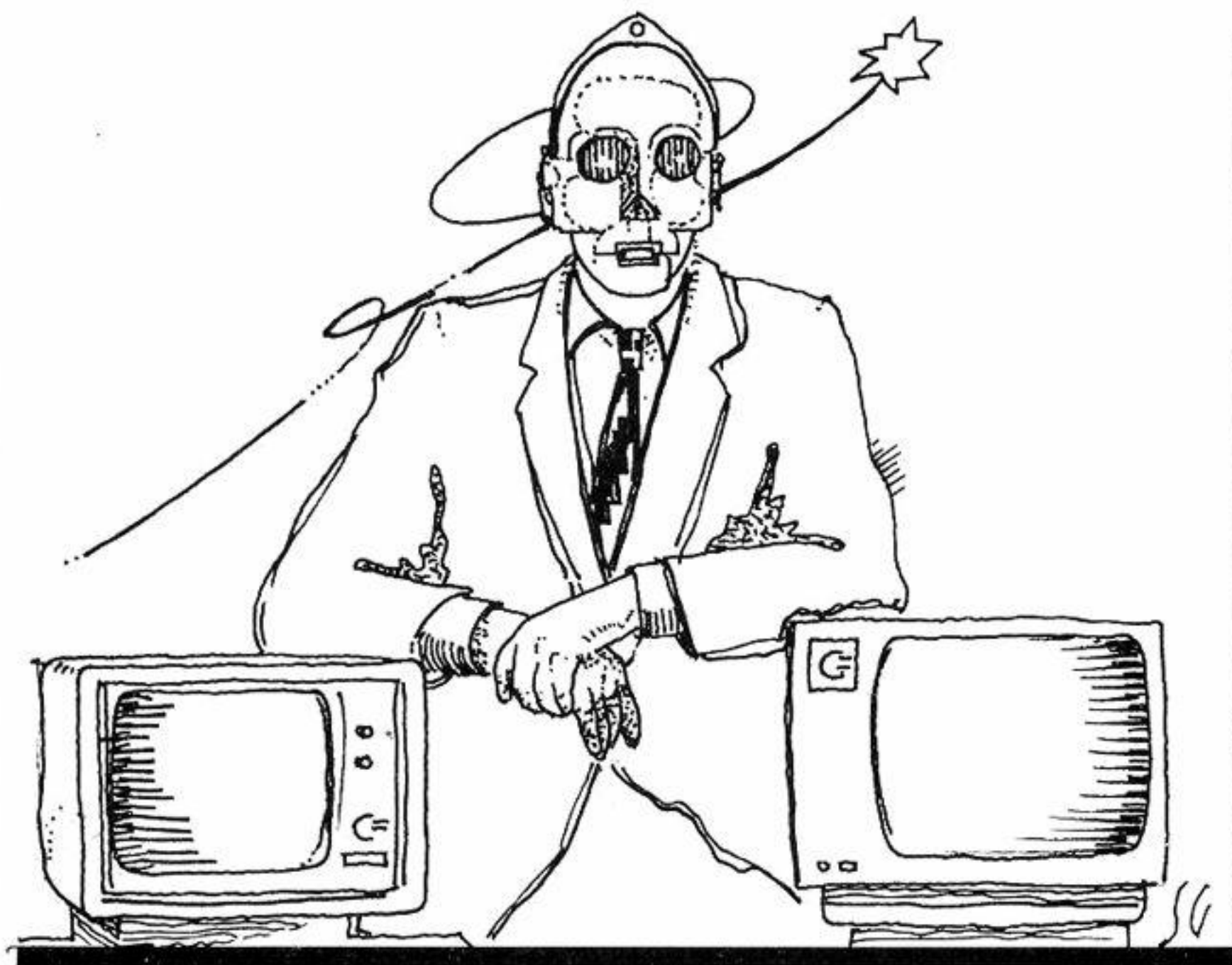
Una volta terminato, il programma in memoria si presenterà come l'originale, per cui sarà possibile listarlo, modificarlo o mandarlo in esecuzione (sia con RUN che con eventuale SYS).

La lunghezza dei programmi da compattare è limitata, in quanto è necessaria la presenza in memoria del programma compattatore.

L'eventuale crash del sistema durante l'esecuzione del programma segnala che l'operazione di "crunch" è stata interrotta a causa della eccessiva lunghezza del programma.

Ai vostri comandi

Una descrizione dettagliata di alcuni potenti comandi disponibili su Amiga

di Luigi Callegari

Abbiamo visto che esiste il comando di ridenominazione dei file RENAME. Ovviamente esiste un comando equivalente che opera con i nomi dei dischetti, o meglio, usando termini precisi, in grado di cambiare il nome associato ad un volume su dischetto. Questo comando ha formato seguente:

RELABEL (DRIVE) drive (NAME) nome

Le parole DRIVE e NAME sono opzionali solo se si specificano i parametri nell'ordine indicato, cioè prima il nome del drive (DF1: DF2:...) ove è inserito il dischetto da ridenominare, poi il nuovo nome da assegnare. Il nome del volume deve seguire le consuete regole di Amigados: non più di trenta caratteri esclusi il doppio punto (:) ed il diviso (/) e da specificare tra virgolette se contiene spazi.

Si noti che questo comando non richiede l'inserimento del dischetto, perciò potrebbe operare soltanto sul dischetto correntenemente selezionato dal sistema per la lettura dei comandi, ovvero la nostra copia di lavoro di Workbench.

Per aggirare l'ostacolo i possessori del solo drive interno possono usare una sequenza del tipo:

COPY C/RELABEL TO RAM: (R)
RAM:RELABEL df0:nuovonome (R)
DELETE RAM:RELABEL (R)

Con tale sequenza si rende residente in memoria, ovvero nel RAMdisk, il comando RELABEL; in seguito si deve cambiare il dischetto e digitare la seconda linea che esegue il comando prelevandolo dal RAMdisk, come specificato dai primi quattro caratteri del nome del comando (RAM:).

Infine si re-inserisce il disco di sistema (Workbench) e si provvede a cancellare il comando, se non più necessario, per rilasciare la memoria. Ovviamente chi possiede anche un drive esterno può inseririvi il disco da ridenominare e, lasciando il disco di sistema nel drive interno, digitare:

RELABEL DF1: nuovonome (R)

Uno dei comandi avanzati ed insoliti, per chi è abituato a vecchi sistemi operativi, ha il seguente formato:

SEARCH (FROM) nome (SEARCH) stringa (ALL)

Permette di ricercare nei file di Amigados una stringa specificata di caratteri. La parola FROM è opzionale, come indicano le parentesi, ma se specificata deve essere seguita dal nome del file o della directory o la maschera di ricerca (vedremo tra poco che cosa sia una maschera Amigados). Il parametro "stringa" è la sequenza di caratteri da ricercare, racchiusa tra virgolette se contiene spazi; la parola SEARCH è opzionale solo se questo parametro segue il "nome". Quando "nome" rappresenta una directory di Amigados, speci-

ficando ALL in coda al comando si ottiene la ricerca della stringa all'interno di tutte le eventuali sub-directory di quella specificata e di tutti i file contenuti. Esempi:

SEARCH df0: Loredana (R)
SEARCH FROM df1:c Mariella ALL (R)

Il primo comando ricerca la stringa "Loredana" nella directory principale (root-directory, convenzionalmente chiamata SYS:) del dischetto inserito nel drive interno. Il secondo comando ricerca la stringa "Mariella" nella directory "c" presente nella directory principale del dischetto nel primo drive interno ed in tutte le eventuali sub-directory, come specifica ALL.

Parlavamo di "maschera" di ricerca. I più esperti, anche i vecchi possessori di C/64 o di MS-DOS avranno intuito che si tratta della solita "pattern matching". Sotto Amigados è possibile cioè indicare con un'apposita stringa di caratteri che un comando deve operare su di un insieme di file i cui nomi hanno dei caratteri in comune.

Se, sotto Amigados, viene ottenuto il nome di una directory dalla maschera, allora il comando può (se gli è concesso dalla sua sintassi interna) operare anche sulla directory e/o sui suoi contenuti. Nei sistemi operativi solitamente si usa un asterisco per indicare caratteri, mentre Amigados usa alcune combinazioni di caratteri speciali: il punto interrogativo indica un carattere qualsiasi, il cancelletto (o crash) seguito da una qualunque maschera indica qualunque o nessuna ricorrenza con quella maschera. Usati insieme, (#?) indicano dunque "un numero qualunque di caratteri qualunque". Inoltre esiste il simbolo di percentuale (%) che vale come "stringa nulla" mentre la barra verticale (presente nella tastiera italiana nel tasto subito prima della "freccia-indietro" di cancellazione, in modo maiusocolo) indica la condizione logica "o". Usando le parentesi tonde si possono raggruppare più maschere. Vediamo qualche esempio:

TYPE pacco?? (R)
DELETE #?.DOC (R)
LIST riel#(6((barra verticale))Z) (R)
DELETE commi (S(barra verticale)(%))#q (R)

Il primo comando stampa su video tutti i file i cui nomi iniziano con "pacco" seguito da altri due caratteri qualsiasi.

Il secondo cancella tutti i file i cui nomi siano composti da qualunque numero di caratteri ma terminanti con ".DOC".

Il terzo comando lista su video tutti i file i cui nomi iniziano con la stringa "riel" e terminanti con un carattere "6" oppure "Z".

Il quarto comando cancella tutti i file i cui nomi iniziano con "commi" che abbiano, o meno, come terzo carattere del nome "S" e che terminano con un numero qualunque di caratteri "q".

Altro comando insolito nei sistemi operativi "vecchi" è quello che permette di riordinare alfabeticamente il contenuto di un file contenente caratteri ASCII. Ricordiamo che i file ASCII con Amiga possono essere generati usando, da CLI, gli editor ED ed EDIT oppure, accessibile anche da Workbench, il sofisticato NOTEPAD. Il formato è:

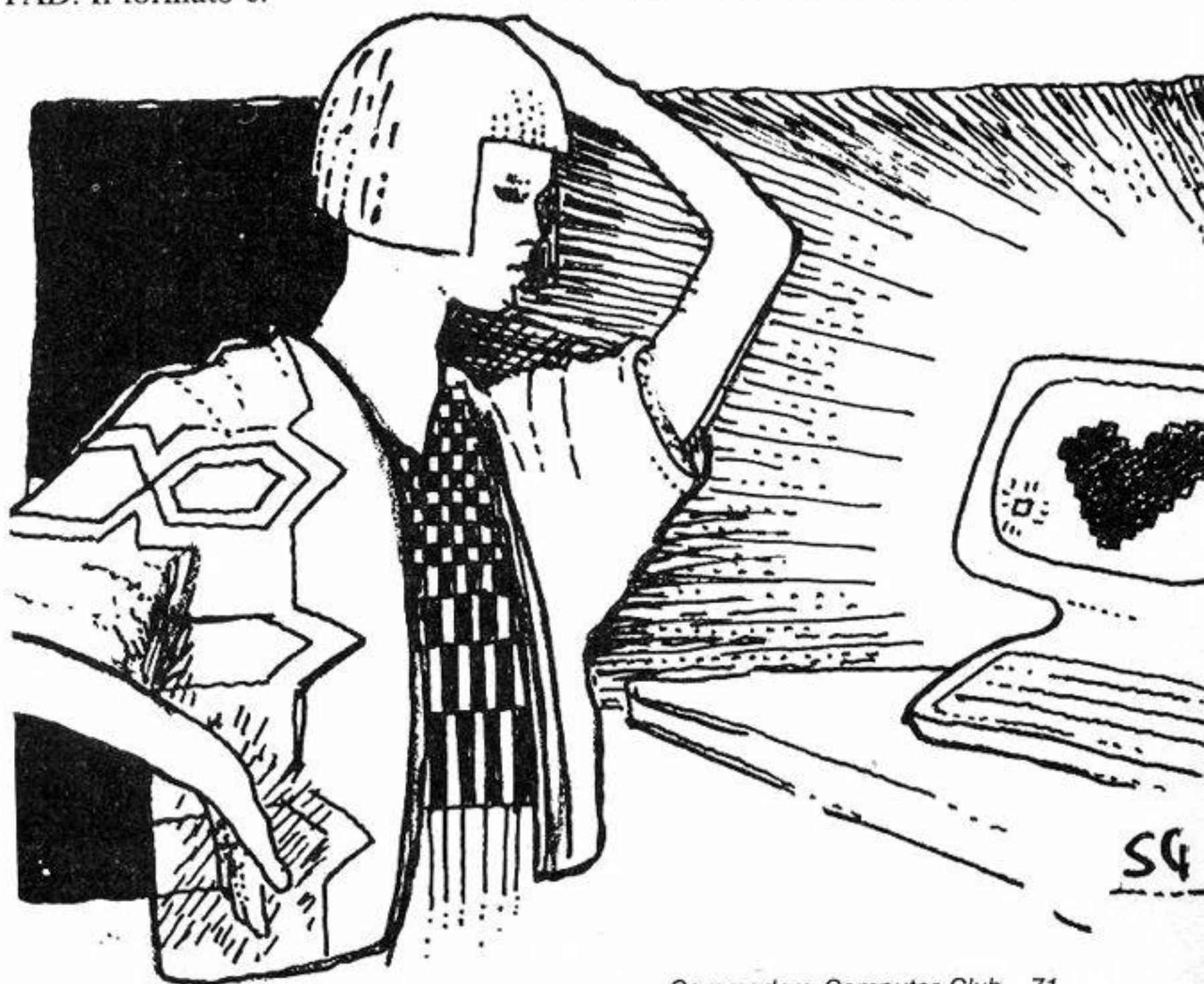
SORT (FROM) origine (TO) destinaz (COLSTART colon)

in cui "origine" è il nome del file da riordinare, "destinaz" è il nome del file ove viene inserito il contenuto del file "origine" riordinato. Come sempre, se il file dove debbono essere inviati i dati non esiste, viene creato appositamente mentre, se già esiste, il suo contenuto viene sostituito dai dati inviati dalla funzione.

Le parole FROM e TO sono opzionali solo se si specificano i parametri nell'ordine indicato, come sempre. Specificando la parola opzionale COLSTART si deve dare un parametro "colon" numerico indicante il punto da cui, per ogni linea, va effettuato il confronto.

Si noti che la funzione SORT, e molte altre di Amigados, prevedono che una linea sia chiusa da un codice ASCII di "LINE-FEED" il cui codice numerico (10) viene inserito dai vari editori ED, EDIT, NOTEPAD quando si preme il tasto Return. Esempi:

SORT confuso TO ordinato COLSTART 9(R)

Legge il contenuto del file "confuso" e lo inserisce, riordinato, nel file

"ordinato" eseguendo i confronti tra le linee partendo dal nono carattere. Ciò significa che se nel file "confuso" si trovano tre linee ASCII come le seguenti...

Loredana Lorenzin
Luisella Dalmazi
Mariella Cavestri

...il sistema considererebbe ciascuna linea partendo dal nono carattere ed ignorando completamente i primi otto. Nel nostro caso considererebbe i cognomi di donna, inserendo nel file riordinato, nell'ordine:

Mariella Cavestri
Luisella Dalmazi
Loredana Lorenzin

Purtroppo i comandi come SORT, che fanno uso intensivo dell'area di memoria riservata al microprocessore 68000 (stack), possono anche bloccare la macchina. Infatti uno dei più gravi handicap del sistema operativo di Amiga consiste nel cattivo controllo delle aree di memoria. In particolare, quando si giunge vicino all'esaurimento della memoria (anche con programmi commerciali come "Deluxe Paint" e "Deluxe music construction set") il sistema non segnala alcun messaggio di avvertimento (tipo "Out of memory") ma si blocca completamente.

Anche operazioni come SORT, che possono esaurire la memoria di lavoro del processore, possono quindi bloccare il sistema. Allo scopo di evitare tale tragico incidente, esiste per fortuna un comando di formato:

STACK (n)

che, specificato senza parametro numerico, visualizza l'attuale quantità di memoria riservata allo stack del processore, di norma 4000 bytes.

Prima di usare SORT con file molto ampi o programmi commerciali sofisticati eseguiti da CLI (e in certi casi, perfino l'interprete BASIC) sarebbe opportuno aumentare notevolmente questo spazio di memoria, per (tentare di) evitare gli incidenti sopra detti.

Per esempio 10000 bytes di stack, solitamente sovrabbondanti per molte applicazioni, si ottengono con:

STACK 10000 (R)

Altro comando che permette di intervenire "in profondità" nella struttura e nel funzionamento di Amigados è il seguente:

ADDBUFFERS drive:nn

Permette di fissare la quantità di memoria supplementare da assegnare ad un drive come memoria di transito dei dati. Senza entrare in dettagli tecnici, che annoierebbero(!) tutti, aumentando la quantità di memoria riservata ad un drive si aumenta la velocità di lavoro degli stessi, in quanto il sistema può trasferire i dati ad una velocità media superiore.

Solitamente sono riservati 512 bytes ad ogni drive. Il parametro "nn" specificato dopo il numero di drive interessato indica quanti blocchi di 512 bytes ciascuno sono da riservare al drive. Si noti che questa memoria non potrà poi essere recuperata (sino a spengimento del computer!). Ad esempio:

ADDBUFFERS df1:11 (R)

riserva (512x11=) 5632 bytes per la gestione dei dati verso e dal primo drive esterno.

Uno dei comandi più complessi di Amigados, permette di conoscere lo stato del "task", ovvero del programma, attualmente in esecuzione. Come si sarà capito, servirà solo ai programmatori più evoluti, ma lo descriviamo egualmente per dovere di cronaca. Presenta il seguente formato:

STATUS (numerotask) (FULL) (TCB) (SEGS) (CLI o ALL)

in cui "numerotask" rappresenta il numero della finestra CLI di cui si vuole sapere lo stato. In difetto, (default) si assume che debbono essere considerati tutti i task attivi.

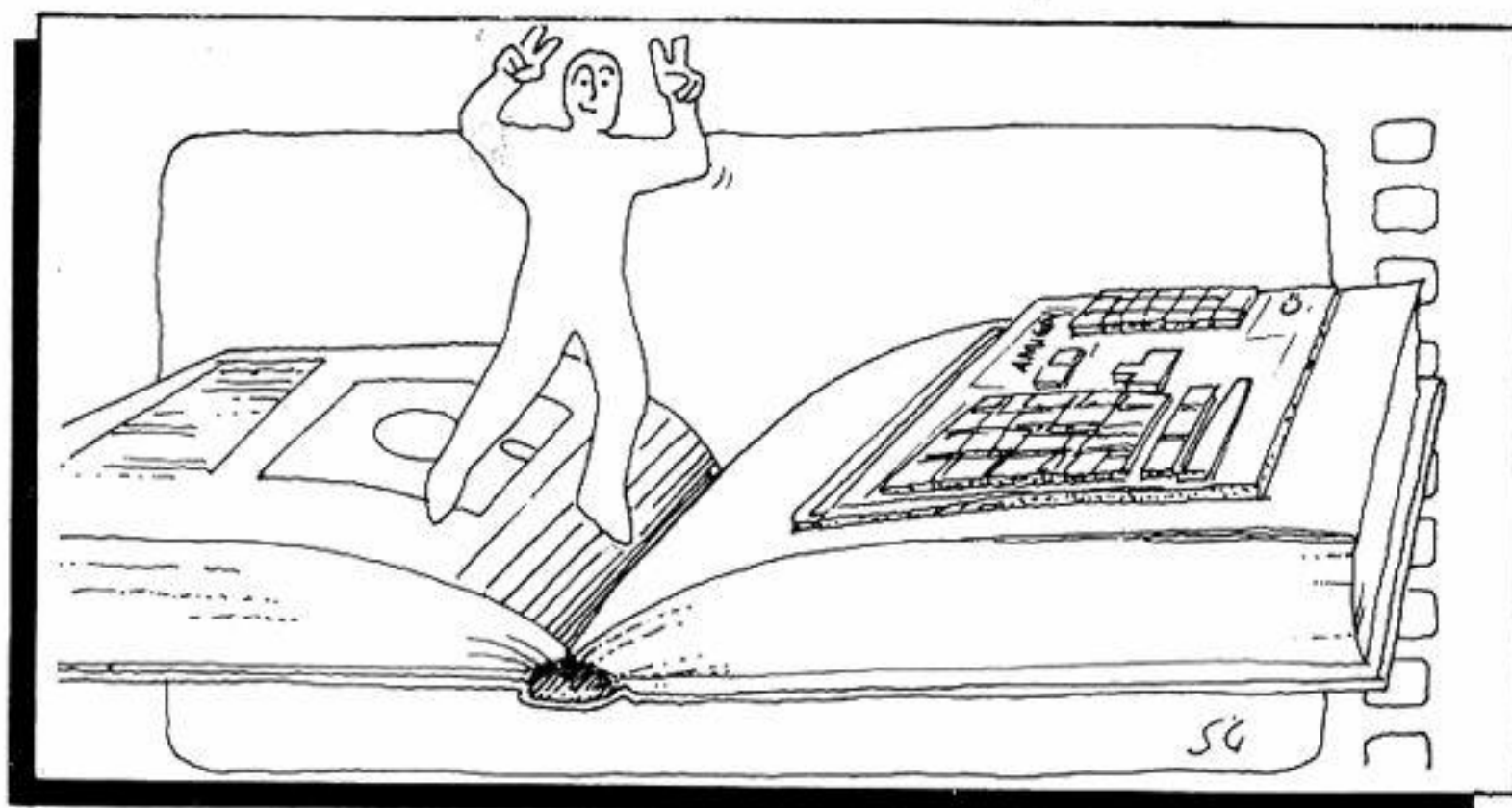
Specificando opzionalmente FULL vengono visualizzate tutte le informazioni possibili, da richiedere singolarmente con le altre parole TCB e SEGS. Questa parola è comunque completamente inutile, perchè, in difetto, il sistema illustra comunque tutti i dati.

Specificando l'opzione TCB si ottiene la visualizzazione delle informazioni riguardanti: quantità di memoria riservata allo stack, dimensione del vettore globale, livello di priorità.

Specificando l'opzione SEGS viene visualizzato il nome della sezione nella lista di segmento.

Specificando l'opzione CLI, oppure l'equivalente ALL, vengono visualizzati solo i nomi delle sezioni di tutti i comandi correntemente caricati nelle finestre CLI correnti (o selezionate).

Come molti avranno capito, la vastità dei comandi di Amigados e la loro potenza impone spesso lunghe digitazioni, specialmente quando si sta lavorando con file molto "ramificati", ovvero inseriti in sub-sub-sub directory.

Esiste un comando che permette di accorciare il lavoro di digitazione. Il suo formato è:

ASSIGN nomeper nomefil (LIST)

Permette di assegnare ad un device (fisico, come una stampante, o logico, come un file) ad una directory.

Il parametro "nomeper" rappresenta il nome del device fisico (RAM: SER: PAR: PRT: NIL: rispettivamente per: ramdisk, interfaccia seriale, interfaccia parallela, interfaccia stampante personalizzata da preferences, fittizio) o directory o drive (DF0: DF1:...).

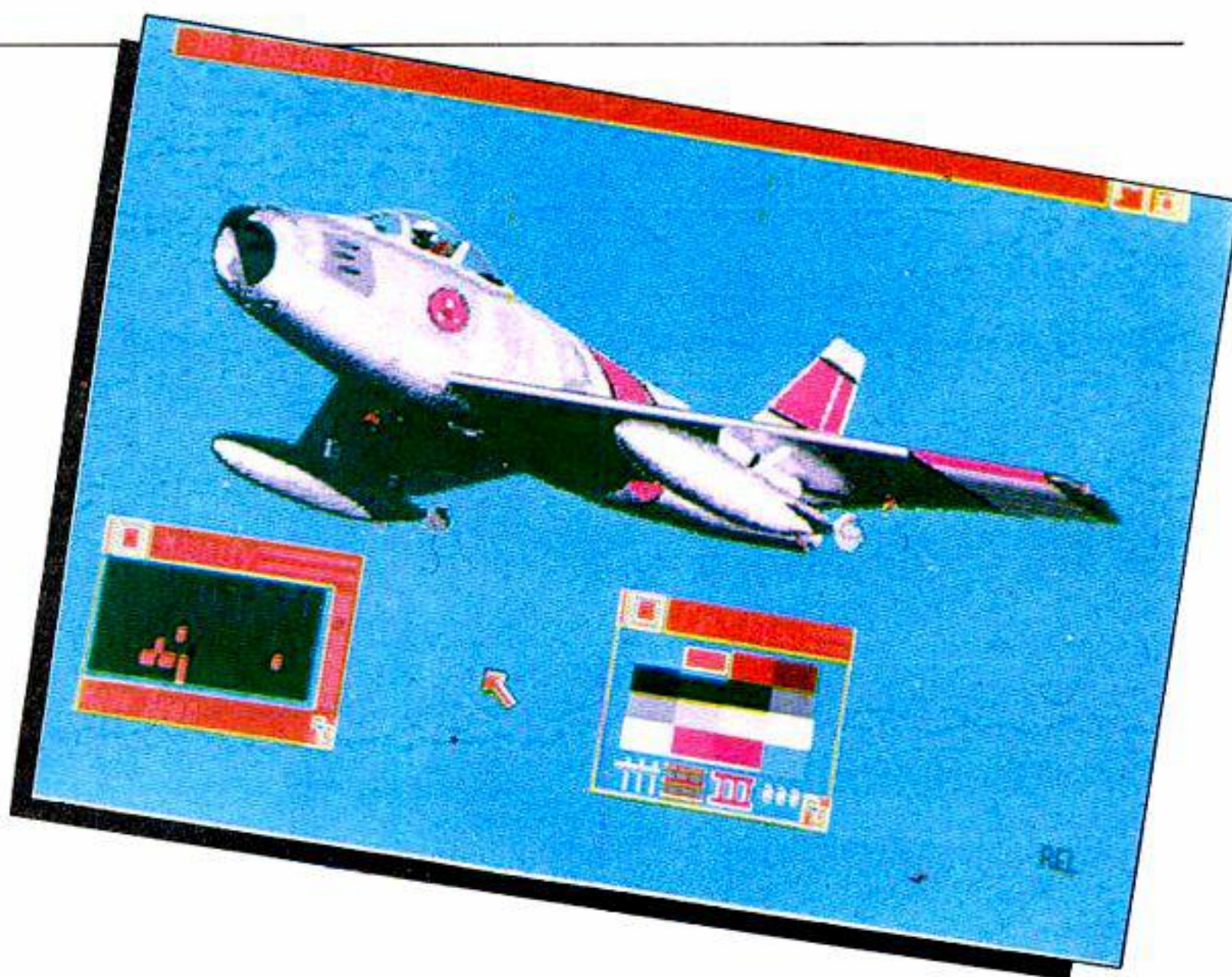
Il parametro "nomefil" rappresenta il nome di disco, il device fisico o la directory che sarà rappresentata dalla specifica "nomeper". Specificando LIST viene visualizzata la lista delle corrispondenze assegnate attualmente. Facciamo qualche esempio:

ASSIGN devs: df1:devs (R)
ASSIGN x: c/LIST (R)

Con il primo comando si indica al sistema che tutte le volte che si digita "devs:" si intende la directory "devs" inserita nella directory madre del dischetto nel primo drive esterno.

Con il secondo comando si indica che tutte le volte che si digita "x:" si vuole eseguire il comando LIST presente nella directory "c".

Restano ora da descrivere solo i comandi da utilizzare nei file "batch" di comandi. E' opportuno, però, descrivere dapprima il modo in cui avviene la redirezione di Input/Output sotto Amigados.

Senza usare troppi paroloni, possiamo dire che normalmente l'input, cioè la lettura di dati, avviene da tastiera mentre l'output, cioè la scrittura ovvero la trasmissione dei dati, avviene sul video in ambiente CLI. Esistono però due caratteri; il "maggiore" (〉7 ed il "minore" (〈) che permettono di specificare, rispettivamente, dove vogliamo mandare, o da dove ricevere, i dati. Come abbiamo già detto, esistono i seguenti nominativi per periferiche (devices):
"RAM:" per il RAMdisk (sezione di memoria usata come un disco)
"SER:" per l'interfaccia seriale RS-232
"PAR:" per l'interfaccia parallela tipo Centronics
"PRT:" per l'interfaccia stampante personalizzata
"NIL:" per l'interfaccia fittizia

Per quanto riguarda le tre interfacce stampanti si noti che i parametri (velocità di trasmissione, bit di parità, modulo continuo/carta singola, alta velocità/Alta qualità eccetera) possono essere regolati solo usando il programma Preferences (anche da Workbench). Questo programma permette anche di fissare altri parametri, come il numero di colonne sul video (60/80), la velocità del cursore, i colori, eccetera e può essere invocato da CLI con un semplice:

System/Preferences (R)

L'uso di Preferences è molto intuitivo: basta usare il mouse e la descrizione del manuale fornito con la macchina. Nel nostro caso, per la regolazione dei device usabili anche da CLI, ci si ricordi di opzionare alla fine del lavoro, in Preferences, "SAVE" per uscire (vicino a "USE"). In questo modo tutti i parametri scelti, dai colori del video ai parametri di controllo delle stampanti (si noterà che sono già previsti "drive" per molte stampanti commerciali) verranno memorizzati dal sistema sul nostro dischetto Workbench o DOS di lavoro (precisamente nel file "system-configuration" nella directory "devs") e saranno automaticamente usati ogni volta che useremo CLI con questo dischetto.

Tornando ai nostri operatori di redirezione, ciò significa che volendo inviare, o meglio "redirigere", l'uscita del comando LIST su stampante seriale (o modem, dipende da che cosa abbiamo collegato al connettore) invece che sul video, si userà semplicemente:

LIST 〉 SER: (R)

Sul video non comparirà niente di LIST, ma la nostra stampante seriale stamperà prontamente i risultati di LIST. Per fare un altro esempio di redirezione di uscita, volendo inviare su stampante il contenuto di un file ASCII chiamato "riel" (posto nella directory "s" nel primo dischetto esterno) si userà:

TYPE 〉 PRT:s/riel (R)

Ovviamente è possibile usare un file, invece di una periferica, per ricevere od inviare dati. Ad esempio, vo-

lendo memorizzare la data corrente e l'ora come prodotti da DATE in un file chiamato "dataodierna", si utilizzerà un semplice:

DATE > dataodierna (R)

Analogamente è possibile inserire in un file ASCII, chiamato "bufdata", tutti i comandi richiesti dal comando date, usando un editor (ED nel nostro esempio) così:

ED bufdata (R)
19-may-87 18:00:00 (R)
(ESCAPE) X (RETURN)

...e fissare automaticamente con DATE quella data (ed ora) usando da CLI:

DATE < bufdata

In questo caso abbiamo rediretto l'input del comando DATE deviandolo dalla tastiera (normalmente assunta come periferica di ingresso in difetto di operatori di redirezione) al file da noi creato.

Il RAMdisk può essere usato esattamente come un dischetto, come abbiamo già visto in articoli precedenti, ma molto più veloce.

In Amigados, versione 1.2, è previsto un apposito comando dal formato:

DISKDOCTOR DRIVE DFn:

In pratica, per tentare il recupero di un dischetto, si deve digitare un comando tipo:

DISKDOCTOR DRIVE df0: (R)

poi attendere che il sistema carichi da Workbench il comando e segnali di inserire nel drive specificato (nel nostro esempio, quello interno) il dischetto da analizzare. Quindi si preme Return.

Il sistema leggerà l'intero contenuto del disco e cercherà di recuperare il maggior numero di file possibile. Sul video, durante il lavoro, vengono segnalati eventuali messaggi per indicare che è stato riscontrato un errore hardware...

"Hard error surface" NN

...oppure software:

"File is unreadable o bad checksum"

Nel primo caso si intende che il difetto è proprio nel supporto del dischetto, perciò alla fine del recupero è vivamente consigliabile gettarlo in pattumiera e comprare in futuro marche migliori.

Nel secondo caso il sistema richiederà se deve cancellare il file oppure lasciarlo inalterato. In seguito viene suggerito, da video, di effettuare una copia del dischetto (DISKCOPY) per recuperare i file ricostruiti e provare a riformattare il disco stesso.

Un dottore per Amiga

Tutti coloro i quali hanno lavorato con vari computer, di qualunque fascia di prezzo, sanno che capita spesso di ritrovarsi con dischetti illeggibili. Non che sia colpa del drive di Amiga, tecnologicamente e tecnicamente uno dei più avanzati nell'ambiente del 3,5' (si pensi che è a doppia testina ed in grado di leggere una traccia intera in un colpo solo) ma spesso capita egualmente che un dischetto male inserito, la mancanza di corrente od una variazione di tensione a causa di un temporale oppure, più frequentemente, dischetti di cattiva qualità, diano questi difetti.

Con gli altri computer, solitamente, si deve ricorrere ad appositi programmi per cercare di recuperare i dati preziosi contenuti nel dischetto difettoso.

A proposito di dischetti, con Amiga è possibile montare anche dei drive per il formato classico 5,25' ormai in via di estinzione pe la maggiore affidabilità dei 3,5. Ciò permette, tra l'altro, di emulare via software o via hardware (leggi Sidecar) rispettivamente il 30% ed il 95% del software MS/DOS (quello degli IBM). Un inconveniente è che il sistema non rileva automaticamente, come per i 3,25, il cambiamento di un dischetto. Esiste perciò, nella versione 1.2, un comando dal formato:

DISKCHANGE DFn:

Esso impone al sistema di leggere i dati relativi al dischetto inserito nel drive da cinque pollici specificato col solito sistema: se è collegato come primo esterno si usa DF1: per il secondo esterno DF2: eccetera.

Si noti che spesso capita di vedere ricomparire file cancellati (DELETE) perchè al momento di impartire il comando di cancellazione non vengono fisicamente cancellati dal disco, ma solo segnalato, con un apposito codice sul primo blocco, che il loro spazio è riutilizzabile, cosa che il sistema può anche non fare subito lasciando praticamente intatto e recuperabile il file.

Perciò DISKDOCTOR permette di recuperare un file sicuramente se è stato cancellato da poco tempo. Questo comando permette di ricostrutire facilmente file grazie alla loro particolare struttura "ad albero doppio" che è fisica sul disco e non solamente logica.

Per i tecnici: non esiste, quindi, un qualcosa di simile alla BAM del C/64 (o di Apple), ma ogni file contiene "l'indirizzo" del precedente e del successivo: partendo da uno solo di essi si può, virtualmente, ricostruire l'intera struttura di un disco.

Tre nuovi comandi per operare in Basic

Come aumentare la potenzialità del vostro C/64 programmando, con una manciata di byte, alcune particolari funzioni

a cura di Alessandro de Simone

FIND (25754/25903)

Find è un'istruzione che si trova spesso nei tool di programmazione in Basic. E' utile durante la correzione di un programma quando è necessario ricercare un'istruzione, o un gruppo di istruzioni, allo scopo di apportarvi modifiche o sostituzioni.

Il comando presentato in queste pagine risulta completo e versatile.

Con la prima delle due forme sintattiche è possibile ricercare solo stringhe, mentre con l'altra si potranno individuare le righe che contengono comandi Basic; abbiamo quindi aggiunto un parametro che permetta di specificare se il gruppo di caratteri da ricercare è un insieme di comandi Basic oppure una stringa. Come certamente saprete, i comandi Basic non vengono memorizzati come sequenza di caratteri ascii, ma sotto forma di codice di un solo byte, detto "Token"; ad esempio, il codice dell'istruzione Print è 153, mentre quello di Goto è 137.

L'operazione di codifica dei comandi Basic viene detta "tokenizzazione" ed ha un duplice scopo: il primo è quello di consentire il risparmio di memoria dal momento che istruzioni che occupano in media 4 byte, saranno contenute in un solo byte; inoltre il programma in esecuzione risulterà più veloce dal momento che non sarà più necessaria l'interpretazione dei comandi, carattere per carattere, ma basterà una tabella in cui siano contenuti i salti da eseguire per le varie istruzioni.

La sintassi per il funzionamento del comando Find è il seguente:

SYS XXXX, 0, "STRING

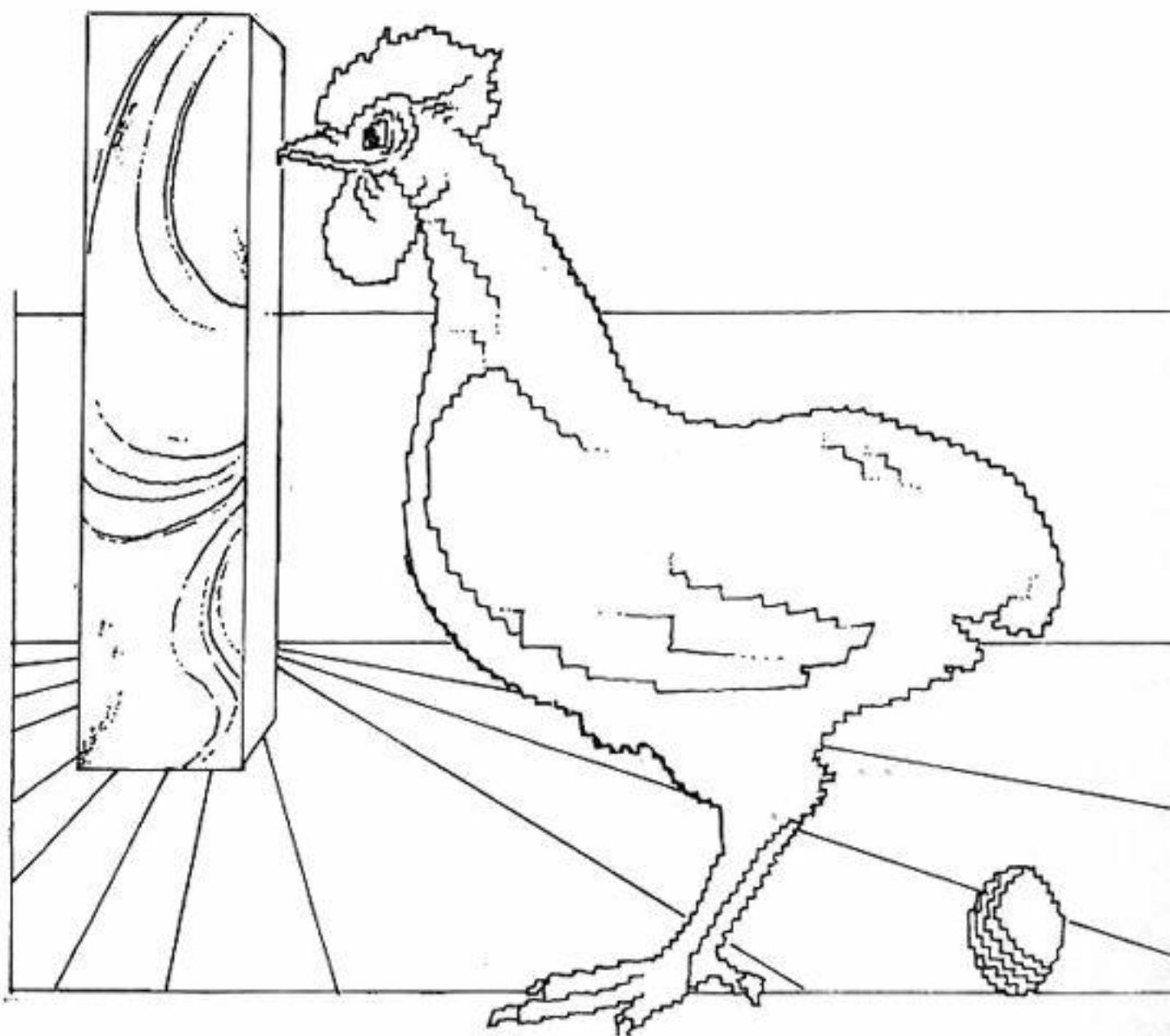
In cui XXXX è l'indirizzo di partenza della routine (interamente rilocabile, come tutte le routine di questa enciclopedia), e "String" è la stringa di caratteri da ricercare; gli apici sono necessari per impedire la tokenizzazione della stringa stessa, che non ne permetterebbe il riconoscimento durante la fase di ricerca.

La seconda forma sintattica è la seguente:

SYS XXXX, OPT, COMM

in cui OPT è un numero qualsiasi compreso tra 1 e 255, e COMM è il comando cercato.

E' possibile inserire più comandi, separati dal carattere doppio punto, purchè sia presente, nel listato oggetto di ricerca, la stessa sequenza.

Non è quindi possibile inserire altre istruzioni di seguito, sulla stessa riga, perchè verrebbero interpretate come comandi da ricercare.

Come utilizzare le routine

Sul N.31 di Commodore Computer Club è iniziata una nuova rubrica che ha lo scopo di venire incontro ai principianti (senza trascurare gli esperti), che desiderano potenziare al massimo le caratteristiche del proprio computer.

Il Basic presenta, infatti, carenze notevoli che possono esser limitate ricorrendo all'uso di routine in linguaggio macchina (LM): è sufficiente attenersi alle istruzioni pubblicate per utilizzare i sottoprogrammi LM con la massima semplicità. Gli "esperti" potranno fare a meno di seguire le istruzioni ed utilizzare direttamente i programmi L.M. pubblicati.

I principianti, invece, è opportuno che leggano con attenzione le "istruzioni per l'uso".

0' Se questa è la prima volta che leggete la rivista, accendete il vostro Commodore 64 e saltate al punto N.2.

1' Accendete il computer e, se desiderate "fondere" alcune (o tutte) le routine di questo numero con quelle tratte dai numeri precedenti (a patto, ovviamente che ne siate in possesso), caricate il file-programma "Nuovo Sistema" (nome standard adottato) con una delle due forme sintattiche che si riferiscono, rispettivamente, ai possessori di nastro o disco:

Load "Nuovo Sistema",1,1
Load "Nuovo Sistema",8,1

Subito dopo digitate NEW e premete il tasto Return.

2' Caricate il programma "Fissa Top di memoria" e lanciatelo col solito RUN. Alla domanda "Ultima locazione?" digitate 20000 e, alla successiva richiesta di conferma, premete il tasto "S". Le altre informazioni che appaiono sul video possono esser comprese solo dagli esperti: i principianti possono tranquillamente ignorarle e saltare alla prossima fase (N.3).

3' Caricate (o digitate dalla rivista) il programma "Caricatore".

4' Digitate dalla rivista la routine che interessa (scritta sempre in Basic, contenente in prevalenza istruzioni Data e numerata da 1000 in poi).

5' Effettuate una copia di sicurezza del programma che rappresenta la "fusione" dei due listati ("Caricatore" + routine Basic pubblicata).

6' Dopo aver digitato Run, alla domanda "Da quale locazione?" rispondete con l'indirizzo iniziale suggerito nello stesso titolo della routine in oggetto. Se il computer, dopo alcuni secondi, visualizza, come indirizzo finale, un valore diverso da quello pubblicato nel titolo (oppure il messaggio "Errore di trascrizione"), interrompete il lavoro (tasti Run Stop e Restore) e verificate con attenzione quanto avete trascritto da rivista.

Se, invece, compaiono messaggi "confortanti" (Routine allocata da... a... Attivare con Sys... ed altre informazioni comprensibili dagli esperti), digitate il programma dimostrativo e lanciatelo: da questo momento avete a disposizione una nuova routine LM da attivare mediante SYS come indicato nelle istruzioni pubblicate per ciascuna routine.

7' Ripetete le operazioni, dal punto 3 in poi, per ciascuna routine pubblicata che intendete

FISSA TOP MEMO

```
100 PRINTCHR$(147)"FISSA TOP DI
     MEMORIA"
110 INPUT "ULTIMA LOCAZIONE";X:
    X-X-1
112 PRINT:PRINT"I VALORI ATTUAL
    I SONO:":PRINT
113 X1-INT(X/256):X2-X-(X1*256)
115 PRINT"PEEK(55):"PEEK(55):PR
    INT"PEEK(56):"PEEK(56)
116 PRINT"FRE(0):"FRE(0)
117 PRINT:PRINT"I VALORI NUOVI
    SAREBBERO:":PRINT:PRINT"PEE
    K(55):"X2
118 PRINT"PEEK(56):"X1
120 PRINT:PRINT"CONFERMI? (S/N)
    "
130 IF PEEK(197)-64 THEN 130
135 IF PEEK(197)<>13 THEN POKE
    198,0: RUN
150 POKE 55,X2:POKE 56,X1: RUN1
    60
160 PRINT"FRE(0):"FRE(0):PRINT:
    PRINT"NEW"
```

SAVE ZONA RAM

```
150 PRINTCHR$(147):INPUT "LOCAZ
    IONE INIZIALE";X
160 INPUT "LOCAZIONE FINALE";T
170 PRINT:PRINT"INIZIO:"X:PRINT
    "FINE:"T
180 IF T<-X THEN  RUN
190 PRINT:PRINT"CONFERMI? (S/N)
    "
200 GET A$:IF A$-"" THEN 200
210 IF A$<>"S" THEN  RUN
220 PRINT:INPUT "NOME FILE";A$
230 PRINT"1- CASSETTA":PRINT"2-
     DISCO"
240 GET B$:IF B$-"" THEN 240
250 IF B$-"1" THEN W-1:GOTO 280
260 IF B$-"2" THEN W-8:GOTO 280
270 GOTO 240
280 PRINTCHR$(147);
290 POKE 198,5:POKE 631,19:POKE
     632,13:POKE 633,13:POKE 63
    4,13:POKE 635,0
300 X1-INT(X/256):X2-X-(X1*256)
    :PRINT"PΓ44,"X1":PΓ43,"X2;
310 Y1-INT(T/256):Y2-T-(Y1*256)
    :PRINT":PΓ46,"Y1":PΓ45,"Y2
320 PRINT:PRINT:PRINT"S*"CHR$(3
    4)A$CHR$(34)","W",1"
330 PRINT:PRINT:PRINT:PRINT:PRI
    NT"SYS64738"
```

CARICATORE

```
150 REM PER UTILIZZARLO, LEGGI
    LE ISTRUZIONI PUBBLICATE SU
160 REM COMMODORE COMPUTER CLUB
170 :
180 Y--1:GOSUB 1000:PRINT:INPUT
     "DA QUALE LOCAZIONE";X
190 READ W:Y-Y+1:IF W<0 THEN 21
    0
200 GOTO 190
210 PRINT"PRIMA LOCAZIONE -"X
220 PRINT"ULTIMA LOCAZIONE -"X+
    Y-1:PRINT
230 PRINT"CONFERMI? (S/N)"
240 GET A$:IF A$-"" THEN 240
250 IF A$-"S" THEN RESTORE :T-X
    :GOTO 280
260  RUN
270 :
280 GOSUB 1000:PRINT:PRINT"ATTE
    NDERE...":PRINT:W-0
290 READ B:IF B>-0 THEN POKE T,
    B:T-T+1:W-W+B:GOTO 290
300 READ B:IF B<>W THEN PRINT:P
    RINTCHR$(18)"ERRORE DI TRAS
    CRIZIONE":END
310 PRINT"ROUTINE ALLOCATA DA"X
    "A"T-1"COMPR.":PRINT
320 PRINT"ATTIVARE CON SYS"X:PR
    INT
330 X1-INT(X/256):X2-X-(X1*256)
    :PRINT"POKE44,"X1":POKE43,"
    X2;
340 Y1-INT(T/256):Y2-T-(Y1*256)
    :PRINT":POKE46,"Y1":POKE45,
    "Y2:END
```

"collezionare" non dimenticando di digitare NEW dopo ogni felice conclusione della fase N.6. Ai principianti consigliamo vivamente di trascriverle tutte in modo da aumentare la propria esperienza e, soprattutto, per evitare incomprensioni degli articoli che leggeranno su Commodore Computer Club.

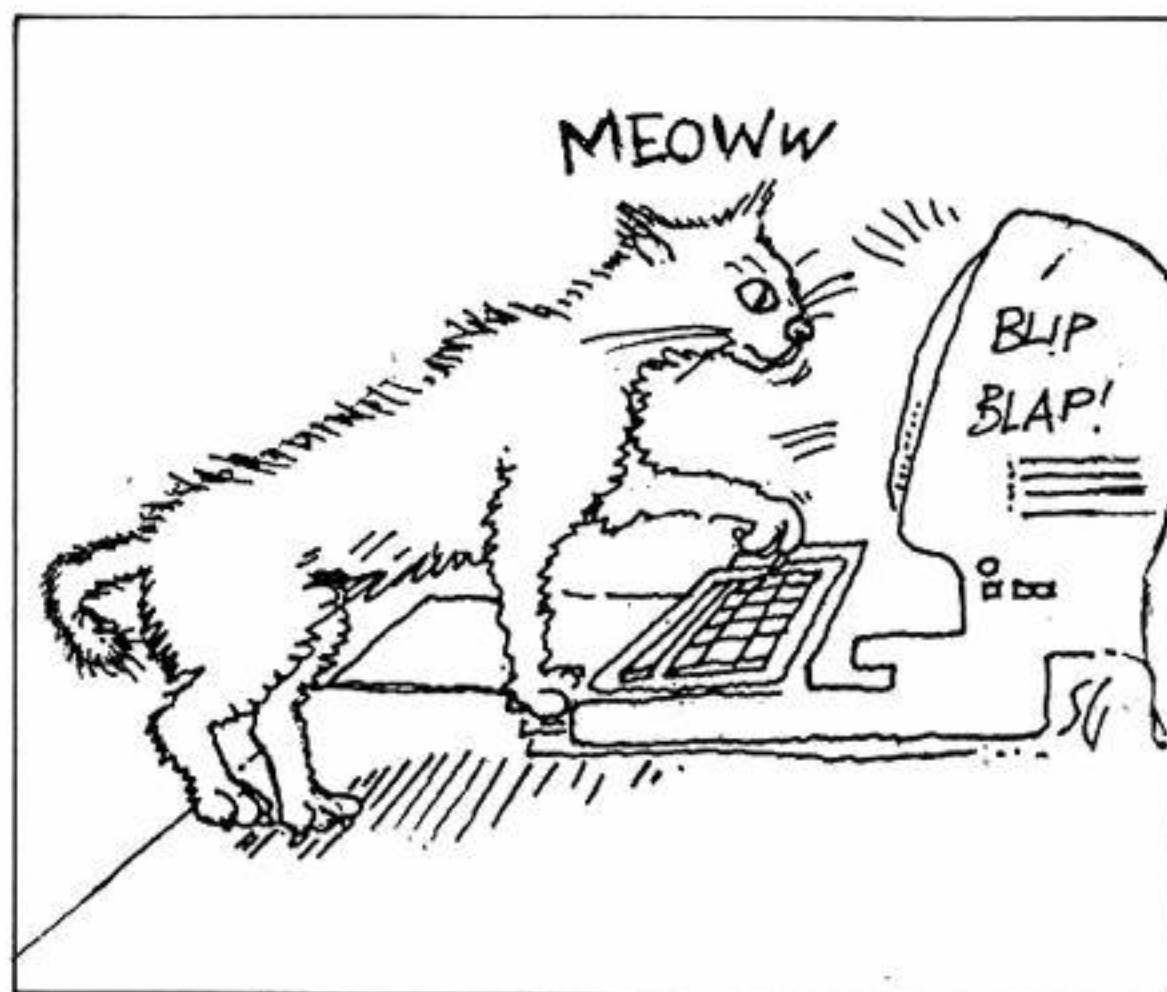
8' Caricate, dopo un nuovo NEW, il programma "Save Zona Ram" e, dopo il Run, alla domanda "Da quale locazione?" rispondete con 20000. Alla seconda domanda "A quale locazione?" ripondete digitando l'indirizzo finale dell'ultima routine trascritta. A seconda se avete un registratore oppure un drive, sul nastro (oppure sul disco) vi ritroverete, dopo aver risposto alle varie domande, il file-programma "Nuovo Sistema" (nome che suggeriamo di assegnare quando compare la relativa domanda). Tale file-programma (da caricare come indicato al punto 1) sarà utilissimo sia per arricchire la vostra raccolta (trascrivendo le routine dei prossimi numeri di Commodore Computer Club), sia per utilizzarle in vostri listati.

9' Digitate SYS 64738 oppure premete il tasto di Reset (se lo possedete) in modo da rimettere "a posto" il computer. Caricate il programma "Fissa Top di memoria" e rispondete con 20000 alla domanda che vi porrà: da questo momento potete disporre sia delle consuete istruzioni Basic che delle routine LM richiamabili con le corrispondenti SYS. Non dimenticate di ripetere la presente fase (N.9) tutte le volte che premete il tasto di Reset o dopo un reset software (SYS 64738). Se, invece, spegnete il computer, sarà necessario attuare la fase N.1 e N.2 per inserire nuovamente nel calcolatore le nuove routine ed usarle senza pericolo.

Collaborazione dei lettori

I lettori che intendono collaborare devono inviare (almeno) tre routine, relativi listati dimostrativi ed articoli esplicativi. Le norme da seguire per la stesura dei listati (piuttosto rigide, per ovvi motivi di compatibilità) sono state segnalate sul N.31. Per ulteriori informazioni, comunque, è possibile telefonare in Redazione (02/8467348) chiedendo di Michele Maggi.

```
1000 PRINTCHR$(147)"FIND"
1010 PRINT:PRINT"SYS XXXX,TYPE,C
     OMLINE"
1020 PRINT:PRINT"TYPE=SPECIFICA
     COMLINE"
1030 PRINT:PRINT"    TYPE=0: COM
     LINE=STRINGA (CHE DEVE"
1040 PRINT"             ESSERE PR
     ECEDUTA DA: ";CHR$(34)
1050 PRINT:PRINT"    1<TYPE<255:
      COMLINE=LISTA COMANDI"
1060 PRINT:PRINT"COMLINE=STRINGA
      O LINEA COMANDI"
1070 PRINT:PRINT"            SE COML
     INE E' UNA STRINGA"
1080 PRINT"            DEVE ESSERE P
     RECEDUTA DA: ";CHR$(34)
1090 RETURN
1100 DATA  173,000,003,141,167,0
     02,173,001,003,141,168,002,
     024
1110 DATA  165,020,105,135,141,0
     00,003,165,021,105,000,141,
     001
1120 DATA  003,032,253,174,032,1
     58,183,138,072,032,253,174,
     104
1130 DATA  208,003,032,115,000,1
     65,043,133,251,165,044,133,
     252
1140 DATA  169,004,133,254,160,0
     00,177,251,200,017,251,208,
     015
1150 DATA  173,167,002,141,000,0
     03,173,168,002,141,001,003,
     076
1160 DATA  059,169,160,000,132,2
     53,164,254,177,251,133,002,
     230
1170 DATA  254,240,227,164,253,1
     77,122,240,014,166,002,240,
     031
1180 DATA  197,002,208,228,230,2
     53,240,209,208,226,160,002,
     177
1190 DATA  251,133,020,200,177,2
     51,133,021,032,019,166,166,
     020
1200 DATA  165,021,076,201,166,1
     60,000,177,251,170,200,177,
     251
1210 DATA  133,252,134,251,024,1
     44,158,-1,19195
```

```
 10 REM  ESEMPIO DI FUNZIONAMEN
    TO FIND
 20 XX=25754:REM  INDIRIZZO SCE
    LTO DA C.C.C.
 25 PRINTCHR$(147)
 30 SYS XX,1,PRINT"PIPPO"
 31 PRINT"LINEA 30"
 35 POKE  198,0:WAIT  198,1:PRI
    NTCHR$(147)
 40 SYS XX,1,"PIPPO"
 41 PRINT"LINEA 40"
 45 POKE  198,0:WAIT  198,1:PRI
    NTCHR$(147)
 50 SYS XX,0,"PIPPO
 51 PRINT:PRINT"LINEA 50"
 55 POKE  198,0:WAIT  198,1:PRI
    NTCHR$(147)
 60 SYS XX,1,PEEK
 61 PRINT"LINEA 60"
 65 POKE  198,0:WAIT  198,1:PRI
    NTCHR$(147)
 70 SYS XX,0,"PEEK
 71 PRINT"LINEA 70"
 80 END
100 PRINT"PIPPO"
110 PRINT"L'ISTRUZIONE PEEK SER
    VE A ..."
120 A=PEEK(1024)
130 REM  PIPPO CHIAMA PIPPO
```

```
Program: FIND
Origin: $C000
By Fabio Sorgato

        LDA $0300 ;Salva gli indirizzi
        STA $02A7 ;della error routine.
        LDA $0301
        STA $02A8
        CLC
        LDA $14   ;Utilizzando l'indiriz-
        ADC #$87  ;zo di chiamata della
        STA $0300 ;SYS, sceglie il
        LDA $15   ;salto della error
        ADC #$00  ;routine.
        STA $0301
        JSR $AEFD ;Prende una virgola
        JSR $B79E ;ed un numero a 8 bit
        TXA       ;e lo salva;
        PHA
        JSR $AEFD ;prende una nuova vir-
        PLA       ;gola e se il parametro
        BNE *C02C ;era 0,
        JSR $0073 ;salta le apici:"
*C02C   LDA $2B   ;Inizia la ricerca
        STA $FB   ;dall'inizio del
        LDA $2C   ;programma
        STA $FC   ;
*C034   LDA #$04  ;Salta il link e il
        STA $FE   ;numero di linea
        LDY #$00  ;se e' la fine del
        LDA ($FB),Y ;programma
        INY
        ORA ($FB),Y
        BNE *C050
*C041   LDA $02A7 ;mette i vecchi punta-
        STA $0300 ;tori alla error
        LDA $02A8 ;routine
        STA $0301
        JMP $A93B ;ed esegue REM
*C050   LDY #$00
        STY $FD
*C054   LDY $FE   ;prende un byte
        LDA ($FB),Y
        STA $02
        INC $FE   ;se la linea e' piu'
        BEQ *C041 ;lunga di 255 caratteri
                  ;torna al basic
        LDY $FD   ;
        LDA ($7A),Y ;se la comparazione
        BEQ *C072 ;ha successo, stampa
                  ;la linea
        LDX $02   ;se la linea basic
        BEQ *C087 ;in esame e' terminata,
                  ;passa alla successiva
        CMP $02   ;se il confronto ha
        BNE *C050 ;successo,
        INC $FD
        BEQ *C041 ;passa alla riga suc-
                  ;cessiva,
        BNE *C054 ;altrimenti ricomincia
                  ;il confronto dal
                  ;primo carattere
*C072   LDY #$02  ;prende il numero
        LDA ($FB),Y ;di linea e lo salva
        STA $14   ;in $14-$15
        INY
        LDA ($FB),Y
        STA $15
        JSR $A613 ;ricerca la linea e
        LDX $14   ;mette l'inizio della
        LDA $15   ;stessa in $5F-$60
        JMP $A6C9 ;stampa la linea;
                  ;la routine termina con
                  ;JMP ($0300), che
                  ;conterra',in questo
                  ;caso $C087
*C087   LDY #$00  ;prende i due bytes
        LDA ($FB),Y ;di link
        TAX       ;e li mette in $FB-FC$,
        INY
        LDA ($FB),Y
        STA $FC
        STX $FB   ;iniziando cosi' a
        CLC       ;scandire una nuova
        BCC *C034 ;linea di programma
```

XOR
(25904/25969)

Il Commodore 64 è provvisto di alcune istruzioni che permettono le operazioni binarie OR, AND e NOT. Un'altra operazione, fondamentale per l'aritmetica binaria, è l'istruzione XOR, assente nel Basic V.2.

L'operazione XOR permette di determinare se due bit sono uguali o meno: se eseguiamo XOR tra due bit uguali il risultato sarà 0, se invece i due bit sono diversi il risulato sarà 1.

Siccome non è possibile la simulazione di XOR tramite OR, AND e NOT (cosa che può avvenire, ad esempio, per il NOR, il NAND ed altri operatori booleani), la routine di queste pagine, lunga una manciata di byte, permette di usare, anche in Basic, l'operatore XOR.

Il "funzionamento" è il seguente:

SYS XXXX, A, B, VAR

in cui, come al solito, XXXX è l'indirizzo di partenza della routine, A e B sono i due numeri (che devono essere compresi tra 0 e 65535), tra cui eseguire l'operazione, e Var è il nome della variabile numerica, in virgola mobile, in cui verrà posto il risulato dell'operazione stessa.

```
1000 PRINTCHR$(147)"XOR"
1010 PRINT:PRINT"SYS XXXX,A,B,NU
     MVAR"
1020 PRINT:PRINT"A & B= I DUE OP
     ERANDI DI: A XOR B"
1030 PRINT:PRINT"       0<=A<=65
     535  0<=B<=65535"
1040 PRINT:PRINT"NUMVAR=VARIABIL
     E NUMERICA CHE CONTERRA'"
```

```
1050 PRINT"          IL RISULTATO D
     ELL'OPERAZIONE"
1060 RETURN
1100 DATA  032,253,174,032,138,1
     73,032,247,183,132,251,133,
     252
1110 DATA  032,253,174,032,138,1
     73,032,247,183,069,252,133,
     252
1120 DATA  152,069,251,133,251,0
     32,253,174,032,139,176,133,
     253
1130 DATA  132,254,165,013,005,0
     14,240,005,162,022,108,000,
     003
1140 DATA  164,251,165,252,032,1
     45,179,166,253,164,254,076,
     212,187,-1,9673
1150 END
```

```
Program: XOR
Origin: $C000
By Fabio Sorgato

       JSR $AEFD ;prende una virgola
       JSR $AD8A ;e un numero intero
       JSR $B7F7 ;positivo a 16 bit,
       STY $FB   ;salva il numero
       STA $FC
       JSR $AEFD ;prende un'altra
       JSR $AD8A ;virgola e un altro
       JSR $B7F7 ;numero
       EOR $FC   ;esegue l'operazione
       STA $FC   ;di or esclusivo (xor)
       TYA
       EOR $FB
       STA $FB

       JSR $AEFD ;prende una terza
       JSR $B08B ;virgola e crea la
                 ;variabile dove mettere
                 ;il risultato
       STA $FD   ;salva i puntatori
       STY $FE   ;alla variabile
       LDA $0D   ;se la variabile non
       ORA $0E   ;e' numerica a virgola
       BEQ *C034 ;mobile

       LDX #$16  ;?TYPE MISMATCH ERROR
       JMP ($0300)
*C034  LDY $FB   ;converte il risulato
       LDA $FC   ;in un numero a virgo-
       JSR $B391 ;la mobile e lo mette
                 :nel flp accumulator

       LDX $FD   ;usando i puntatori,
       LDY $FE   ;trasferisce il flp acc
       JMP $BBD4 ;nella variabile
```

DUMP (25970/26172)

Dump è un'istruzione che non richiede parametri. Serve, infatti, a riportare la lista delle variabili utilizzate fino a quel momento.

Impartita la SYS XXXX, in cui XXXX è la locazione di inizio della routine, apparirà la lista delle variabili intere, stringa e in virgola mobile fino a quel momento elaborate.

Per riportare la lista su stampante è necessario eseguire:

OPEN 1,4: CMD 1: SYS XXXX

Al termine della stampa sarà necessario riportare l'output su video con:

PRINT#1: CLOSE 1

```
1000 PRINTCHR$(147)"DUMP"
1010 PRINT:PRINT"SYS XXXX"
1020 PRINT:PRINT"PER OUPUT SU ST
     AMPANTE:"
1030 PRINT:PRINT"   OPEN 1,4:CMD
      1:SYS XXXX"
1040 PRINT:PRINT"   AL TERMINE:P
     RINT#1:CLOSE1"
1050 RETURN
1100 DATA  160,000,165,045,166,0
     46,134,096,133,095,228,048,
     208
1110 DATA  004,197,047,240,070,2
     00,177,095,136,041,128,240,
     065
1120 DATA  177,095,041,128,208,1
     02,177,095,041,127,032,210,
     255
1130 DATA  200,177,095,041,127,0
     32,210,255,169,036,032,210,
     255
1140 DATA  169,061,032,210,255,1
     69,034,032,210,255,160,003,
     177
1150 DATA  095,133,034,200,177,0
     95,133,035,160,002,177,095,
     032
1160 DATA  036,171,169,034,032,2
     10,255,024,144,088,096,144,
     175
1170 DATA  177,095,032,210,255,2
     00,177,095,032,210,255,169,
     061
```

```
1180 DATA  032,210,255,024,165,0
     95,105,002,072,165,096,105,
     000
1190 DATA  168,104,032,162,187,0
     32,221,189,032,135,180,032,
     033
1200 DATA  171,024,144,042,160,0
     00,177,095,041,127,032,210,
     255
1210 DATA  200,177,095,041,127,0
     32,210,255,169,037,032,210,
     255
1220 DATA  169,061,032,210,255,2
     00,177,095,072,200,177,095,
     168
1230 DATA  104,032,145,179,024,1
     44,202,160,000,024,165,095,
     105
1240 DATA  007,133,095,165,096,1
     05,000,133,096,169,013,032,
     210
1250 DATA  255,165,095,166,096,0
     24,144,142,-1,25154
```

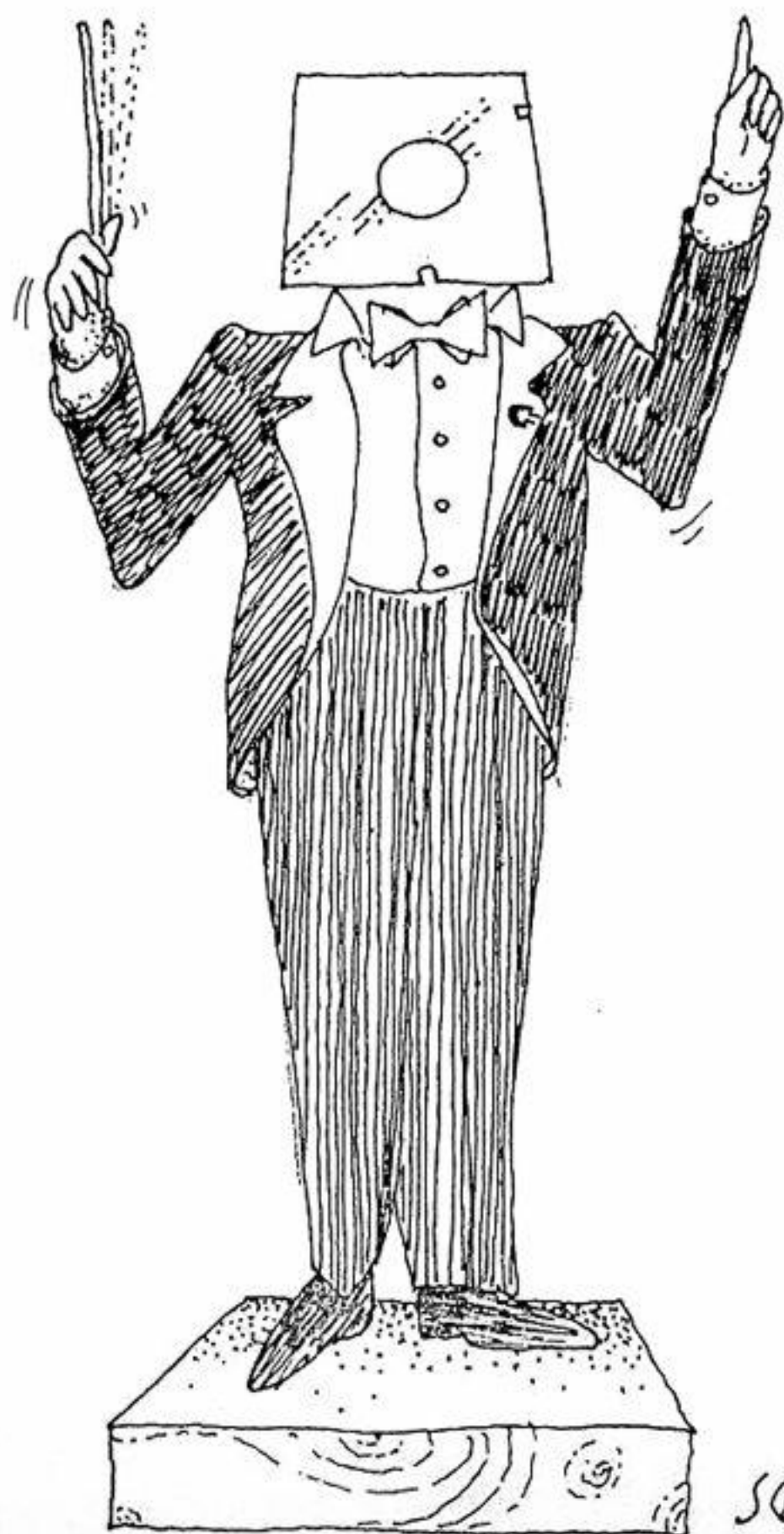
```
Program: DUMP
Origin: $C000
By Fabio Sorgato

      LDY #$00 ;inizia il LIST delle
      LDA $2D  ;variabili dall'inizio
      LDX $2E  ;dell'area riservata
      STX $60  ;alle variabili

      STA $5F
*C00A CPX $30  ;se e' terminata la
      BNE *C012 ;scasione
      CMP $2F
      BEQ *C058 ;torna al basic.
*C012 INY
      LDA ($5F),Y ;prende il secondo
      DEY         ;carattere del nome
      AND #$80    ;della variabile, se il
                  ;bit 7 e' a 0, la
                  ;variabile e' numerica
      BEQ *C05B ;a virgola mobile
      LDA ($5F),Y ;se il bit 7 del
      AND #$80 ;primo carattere e' a 0,
      BNE *C086 ;la variabile e stringa

      LDA ($5F),Y ;stampa il primo
      AND #$7F
      JSR $FFD2
      INY
      LDA ($5F),Y ; e il secondo carat-
      AND #$7F   ;tere del nome,
      JSR $FFD2
      LDA #$24   ;stampa $
      JSR $FFD2
      LDA #$3D   ;stampa =

      JSR $FFD2
      LDA #$22   ;stampa "
      JSR $FFD2
      LDY #$03
      LDA ($5F),Y ;usando i puntatori
      STA $22      ;della variabile
      INY
      LDA ($5F),Y
      STA $23

      LDY #$02    ;e la lunghezza,
      LDA ($5F),Y
      JSR $AB24 ;stampa la variabile
      LDA #$22  ;stampa "
      JSR $FFD2
      CLC        ;ricerca la prossima
      BCC *C0B0 ;variabile.
*C058 RTS        ;Ritorno al basic
*C059 BCC *C00A ;Richiamo di salto.
*C05B LDA ($5F),Y ;stampa il primo
      JSR $FFD2
      INY
      LDA ($5F),Y ;e il secondo
      JSR $FFD2    ;carattere del nome,
      LDA #$3D     ;stampa =
      JSR $FFD2
      CLC
      LDA $5F     ;usando i puntatori
```

```
       ADC #$02  ;alla variabile,
       PHA
       LDA $60
       ADC #$00
       TAY
       PLA
       JSR $BBA2 ;carica il contenuto
                 ;della variabile
                 ;nel flp accumulator
*C07A  JSR $BDDD ;converte in stringa
       JSR $B487 ;prepara i puntatori
                 ;di stampa
       JSR $AB21 ;e stampa la variabile
       CLC       ;passa alla variabile
       BCC *C0B0 ;successiva
*C086  LDY #$00  ;
       LDA ($5F),Y ;stampa il primo
       AND #$7F
       JSR $FFD2
       INY
       LDA ($5F),Y ;e il secondo
       AND #$7F    ;carattere del nome
       JSR $FFD2
       LDA #$25    ;stampa %
       JSR $FFD2
       LDA #$3D    ;stampa =
       JSR $FFD2
       INY
       LDA ($5F),Y ;prende il byte
       PHA         ;alto e
       INY
       LDA ($5F),Y ;il byte basso,
       TAY
       PLA         ;converte in
       JSR $B391   ;floating point (flp)
       CLC         ;e stampa come
       BCC *C07A   ;se fosse una varia-
                   ;bile flp
*C0B0  LDY #$00    ;passa alla variabile
       CLC         ;successiva,
       LDA $5F
       ADC #$07    ;saltando i 7 bytes
       STA $5F     ;di cui e' composta
       LDA $60     :una variabile
       ADC #$00
       STA $60
       LDA #$0D    ;stampa un CR
       JSR $FFD2
       LDA $5F
       LDX $60
       CLC
       BCC *C059   ;e continua
```

(Le routine di questo numero sono opera di Fabio Sorgato).

Tecniche di Auto Run per C/128

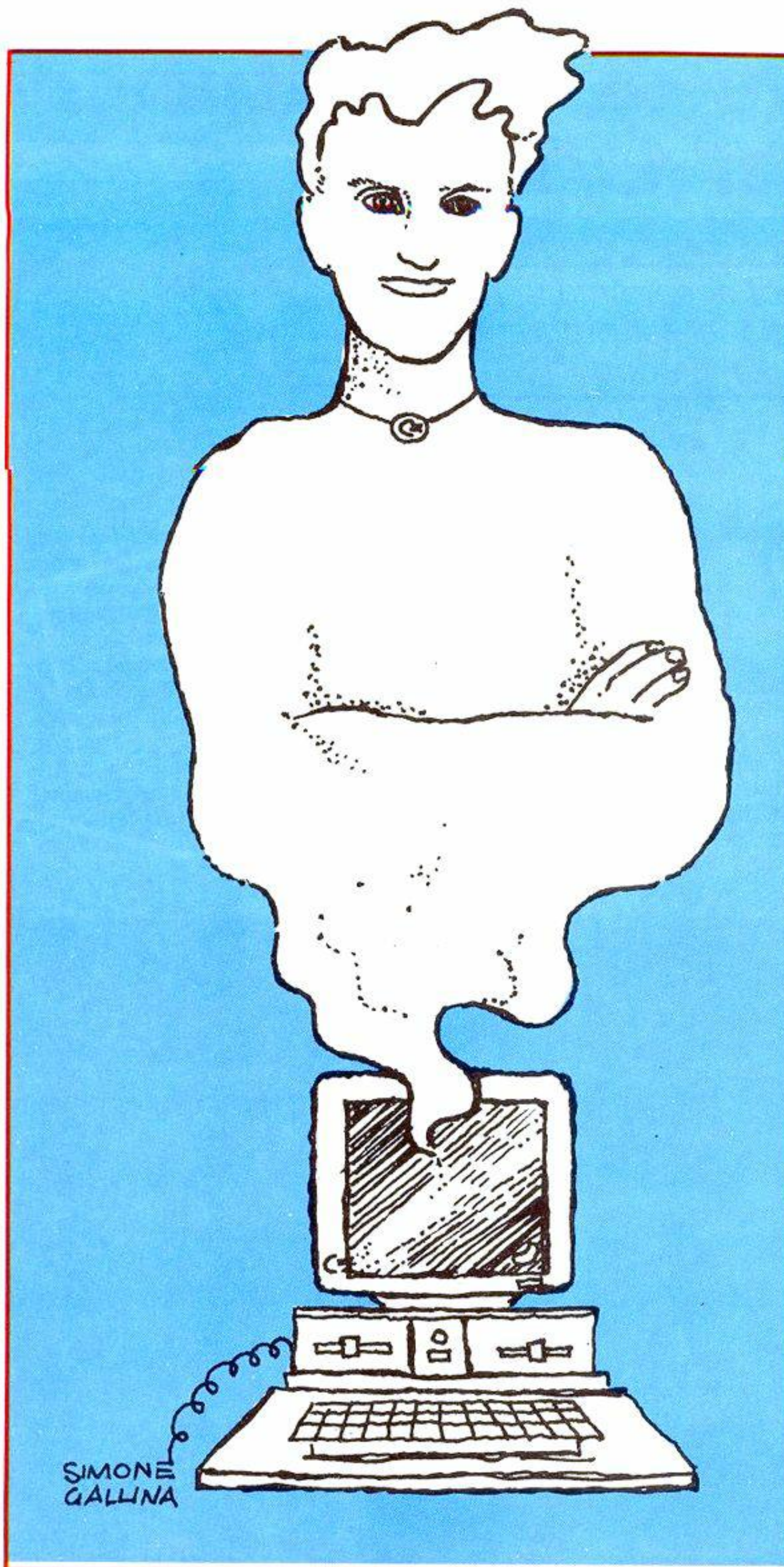
Un paio di accorgimenti per tentare la protezione dei programmi scritti in "modo" 128

di Domenico Pavone

Una delle caratteristiche che rendono evidente il salto qualitativo dal C/64 al C/128, è senza dubbio la possibilità di far partire un programma semplicemente inserendo un dischetto nel drive e accendendo il computer (procedura di boot strap).

Pur essendo ovvi i grossi vantaggi che si possono ottenere con tale tecnica, è altresì evidente che se un dischetto è "imbottito" dei 144 files che può contenere, soltanto uno potrà godere del privilegio di essere caricato e lanciato all'accensione o tramite il comando BOOT. Si può anche utilizzare la pressione contemporanea dei tasti Shift e Run/Stop per lanciare il primo programma presente nella directory, ma il problema non si discosta di molto.

Quindi, a meno di non utilizzare un dischetto per ogni programma, un buon "autorun" può risultare utile anche per il C/128 in modo, ovviamente, 128: vuoi per rendere (un po') più difficile la sprotezione di un nostro programma, vuoi per risparmiare l'immane fatica di digitare RUN dopo il caricamento.

Il tema dell'autostart (anche se con riferimento ad altre macchine) è stato trattato svariate volte su questa rivista. Dovreste già sapere che il "dirottamento" di alcuni vettori del sistema operativo è quasi sempre posto alla base della realizzazione di un qualsiasi Run Maker. Entrando

nel merito, ciò che consente ad un programma di partire non appena termina il caricamento, è che invece di eseguire la routine interna di READY, va direttamente a quella di RUN (o ad una nostra routine che si conclude con un salto al RUN).

Tutto questo, nel C/64, significava alterare due locazioni, la $0302 e la $0303 (770-771), che contenevano l'indirizzo del famigerato Basic Warm Start.

Nel C/128, invece, il Basic Start è puntato dalle locazioni...$0302 e $0303! Già, proprio le stesse. Questo però non vuol dire che si possano usare in blocco i Run Maker per C/64 limitandosi a limare qualche differenza di locazione. Nel C/128, infatti, bisogna anche tenere conto di altri fattori, come ad esempio l'inizio dell'area Basic, normalmente posto alla locazione $1C01 (7168) che, se si inizializza la pagina grafica, viene spostato a $4001 (16385).

L'indirizzo di fine programma, inoltre, non coincide con quello di inizio delle variabili (come nel caso del C/64): per queste viene utilizzato il banco 1.

Ma verificheremo più chiaramente queste considerazioni facendo riferimento ai due programmi pubblicati, la cui descrizione semplificherà quanto sinora detto. Le due routines per raggiungere lo stesso obiettivo seguono due metodi differenti, e la loro diversa applicazione potrà risultare più o meno vantaggiosa a seconda delle vostre esigenze. Per chi non voglia addentrarsi nell'approfondimento dela struttura dei due listati, basterà digitarli avendo l'accortezza di inserire due linee tipo...

1 TRAP 1000
1000 PRINT ERR$(ER): HELP: SLOW: END

...che potranno poi essere cancellate una volta raggiunta la certezza che tutto funzioni correttamente. Una volta lanciati, entrambi i programmi fanno tutto da soli.

Anzitutto chiedono quale file si vuole dotare di autorun, il nome da assegnare al file oggetto (quello "lavorato") e se si desidera, o meno, l'inibizione del tasto Run/Stop. Se si

risponde positivamente a quest'ultima domanda, il programma con l'autorun ripartirà automaticamente se si cerca di fermarlo con la pressione di STOP, di STOP+RESTORE e, addiritura, con il tasto Reset.

A questo punto verrà generato sullo stesso dischetto un file con differenti caratteristiche a seconda del MAKER utilizzato. Il primo creerà un loader molto breve (1 blocco) con il nome da voi scelto, che una volta caricato con BLOAD oppure LOAD "nome file",8,1 a sua volta caricherà il programma desiderato e lo manderà in esecuzione.

Il secondo produrrà un file unico comprendente il programma originale, sempre col nome che gli avrete assegnato, da caricare anch'esso nella forma rilocata (BLOAD). Questo file sarà però 25 blocchi più lungo dell'originale, per cui a voi la scelta sul rapporto vantaggio / svantaggio dei due sistemi.

Un'altra considerazione utile ai fini di tale scelta è che il sistema "loader" funziona indipendentemente dalla conformazione in cui si opera rispetto alla pagina grafica, mentre il secondo metodo presenta un inconveniente: volendo, potete creare l'autorun con la pagina grafica impostata (magari per vostre successive esigenze), ma il file generato in questo caso sarà ben 61 blocchi più lungo, in quanto comprenderà anche la memoria da $1C01 a $4000.

Una precisazione quasi superflua: i programmi che si possono trattare devono essere Basic, oppure in LM purchè dotati di una linea tipo "10 SYS...". Inoltre, anche se non indispensabile, è conveniente spegnere e accendere il computer prima di usare i "maker".

Il primo listato

Compito di questo programma è la creazione sul dischetto di un file che all'atto del caricamento si disponga nelle locazioni da $0263 a $0303 (decimale: 611/771), facendo sì che le ultime due (che corrispondono al vettore di Basic Start) puntino all'inizio di una routine posta a partire da $0263.

Per ottenere ciò si apre un file programma in scrittura (col nome determinato dall'input di riga 30) e si inseriscono, come primi, due dati: i valori (ovviamente nel formato basso/alto) che determineranno la locazione a partire dalla quale esso verrà caricato (righe 100-110). Subito dopo viene inserita la routine in linguaggio macchina che si occuperà del caricamento del programma da lanciare (righe 120-160), con in coda i valori ASCII del nome di questo ed eventuali spazi se la lunghezza del nome è inferiore ai 16 caratteri (170-180). A questo punto si immettono nel file su disco i valori presenti da $02A2 a $0301 senza alterarli (riga 190) e infine, in quelle che saranno le locazioni $0302 e $0303, i puntatori a $0263 (riga 200).

Quando il file così composto verrà caricato con BLOAD, in pratica manderà immediatamente in esecuzione la routine che inizia a $0263. Questa , dopo aver disabilitato i messaggi del S.O. e aver pulito lo schermo, provvederà a reinserire nelle locazioni di Basic Start i valori di default o, se si è scelto diversamente (righe 50-80), vi allocherà l'indirizzo della routine di RUN ($5AA6).

Quindi la routine si occupa del caricamento del programma richiesto usando il LOAD del kernal ($FFD5), il quale prevede che se si setta un file tramite la routine posta a $FFBD con un indirizzo secondario uguale a zero (vedi disassemblato da $275 a $27A), il caricamento inizierà dalla locazione puntata dai registri X e Y, nei quali troverà inseriti i valori dei puntatori di inizio Basic ($2D-$2E), aggiornati automaticamente dal S.O. se si setta la pagina grafica.

Effettuato il LOAD, non resta che saltare alla routine di RUN... e il gioco è fatto.

Da tenere presente che il file generato, oltre al vettore di Basic Start include anche quello della routine di errore, posto a $300 e $301 (768 e 769 in dec.). Se si vuole che, in caso di errore nel programma, questo non si fermi, sarà sufficiente modificare la riga 190 in...

FOR X=674 TO 767

...(il resto è uguale) e aggiungere la linea...

195 PRINT#5, CHR$(LOW);: PRINT#5, CHR$(HI);

...(facendo attenzione ai caratteri di punto e virgola) dove LOW e HI sono i puntatori alla routine cui saltare in caso di errore. Ulteriori protezioni potete realizzarle aggiungendo i vettori posti subito dopo il Basic Start, basterà aggiungere in coda alla linea 200 tutte le
PRINT#5,CHR$(xx);
che volete, tenendo presente che i vettori più "interessanti" sono:

$318 $319 :Interruzione non mascherabile
$328 $329 :Stop
$332 $333 :Save

Il secondo listato

Il flusso operativo di questo programma può essere distinto in due fasi: 1) Caricamento del file da trattare. 2) Salvataggio dello stesso includendo la memoria che va da $0300 all'inizio del Basic, dopo averne modificato i vettori di Basic Start.

Allocata in memoria la routine in LM a partire da $02E5 (righe 100-150) ed eventualmente (in base alla risposta all'input di riga 50) la miniroutine a $B00, la prima fase descritta viene eseguita da Basic. Per ottenere lo scopo, prima si caricano due locazioni del buffer di tastiera (nel C/128 situato da 842 a 851 dec.) con il carattere 13, ASCII di RETURN; quindi, dopo aver scritto sullo schermo (con lo stesso colore dello sfondo) in posizione adeguata:

DLOAD "nome file"

...e...

SYS 606

(=$2E5), si impartisce NEW che, oltre a cancellare il programma, provocherà lo svuotamento del buffer di tastiera con conseguente caricamento del programma desiderato e salto alla routine LM. Questa si

occuperà della seconda fase: modificherà i valori di Basic Start in maniera da farli puntare a RUN o, se si è risposto no al quesito delle righe 60-90, li dirotterà ad un'altra routine posta da $0B00 (buffer di cassetta) che verrà salvata assieme al programma da trattare. Per il SAVE vero e proprio si usano le locazioni libere di pagina zero $FA e $FB per segnalare al kernal l'indirizzo da cui iniziare (il valore FA deve essere inserito nell'accumulatore), e che sarà $0300; mentre per la posizione di fine caricamento verranno sfruttate le locazioni $AE ed $AF, nelle quali il S.O. deposita l'indirizzo di fine programma dopo un LOAD. Una volta effettuato il salvataggio, il computer va al RESET di sistema.

La routine posta a $B00 si limita a ripristinare i valori standard di Basic Start, ma nulla vieta che vi inseriate tutte le modifiche che volete prima di saltare a $5AA6 per il run, ponendo in coda al programma basic altre linee DATA e modificando il valore della FOR di riga 160.

Durante il caricamento dei files generati da entrambi i Run Maker lo schermo di 40 colonne risulterà vuoto così come pure quello in 80 colonne usando il sistema "loader", mentre con i programmi in autorun su unico file il video 80 colonne resterà attivo.

Per concludere, non resta che invitarvi ad una certa attenzione usando il secondo Run Maker, in quanto non è fornito di controllo sugli errori del D.O.S. Un eventuale FILE EXISTS o FILE NOT FOUND vi porterà a dover ricominciare da capo.

SCHEDA TECNICA

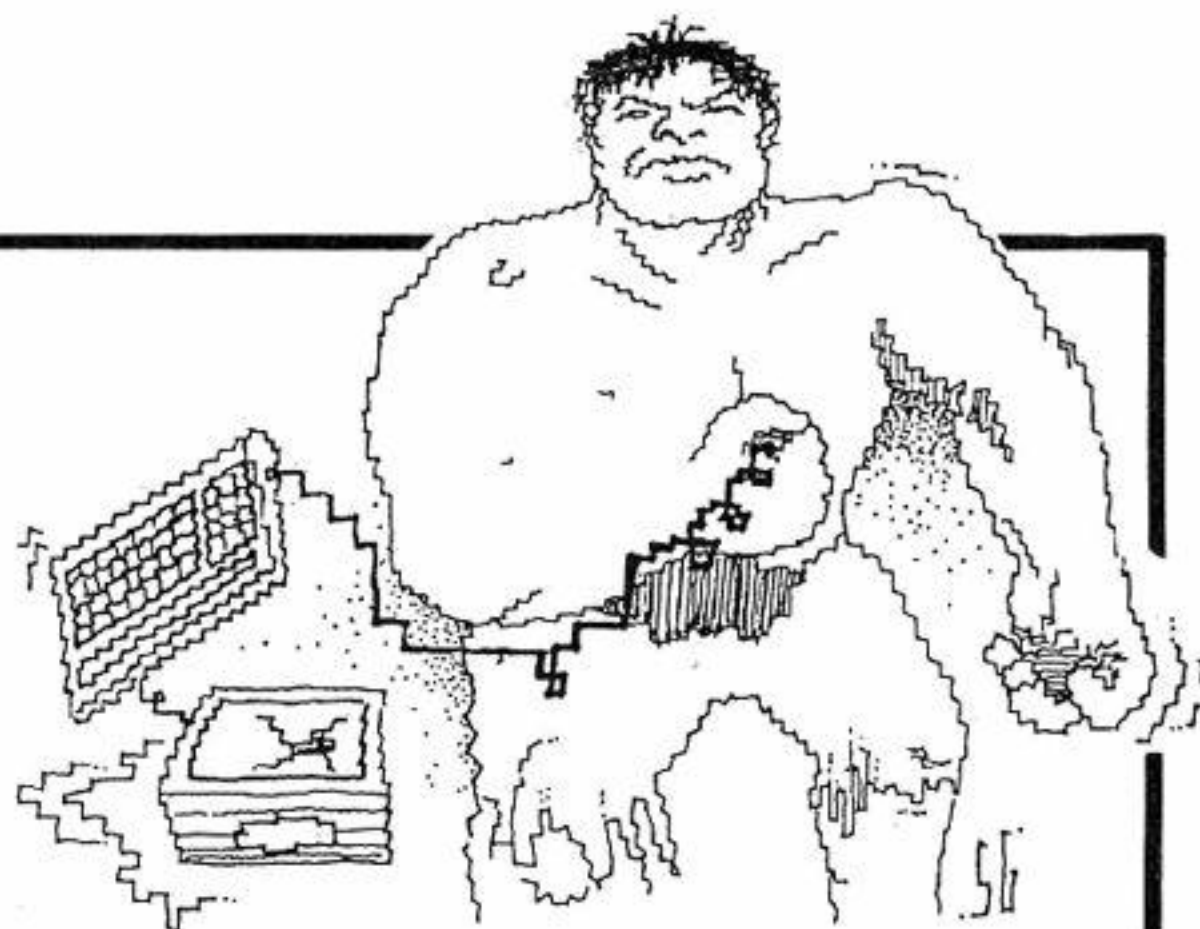
Software applicativo per protezioni di programmi.

Idoneo per computer C/128 e non adattabile ad altri computer Commodore

Richiede il disk drive

Anche i programmi pubblicati in queste pagine sono contenuti nel disco "Directory" di questo mese.

```
3 REM COMMODORE 128 + DRIVE
4 :
5 REM RUN MAKER TRAMITE LOADER
6 REM BY DOMENICO PAVONE
7 :
10 SCNCLR:COLOR0,1:COLOR4,1:COLOR6,1:COLOR5,16
20 INPUT"NOME FILE DA CARICARE";NF$:LF=LEN(NF$):IFLF>16ORLF<1THEN10
30 SCNCLR:INPUT"NOME DA ASSEGNARE AL LOADER";NL$:LL=LEN(NL$):IFLL>16ORLL<1THEN30
40 SCNCLR:PRINT"INIBIZIONE DELLO STOP (S/N)?"
50 GETKEYX$
60 IFX$="S"THENLI=166:HI=90:REM PUNTATORI A 'RUN'
70 IFX$="N"THENLI=198:HI=77:REM PUNTATORI DI DEFAULT
80 IFX$<>"S"ANDX$<>"N"THEN50
90 FAST:SCNCLR:PRINT"ELABORAZIONE IN CORSO"
100 OPEN5,8,5,NL$+",P,W":IFDS<>0THENSCNCLR:PRINTDS$:CLOSE5:SLOW:END
110 PRINT#5,CHR$(99);:PRINT#5,CHR$(2);:REM INIZIO CARICAMENTO DA $0263 (611)
120 FORX=611TO619:READA:PRINT#5,CHR$(A);:NEXT
130 PRINT#5,CHR$(LI);:PRINT#5,CHR$(162);:PRINT#5,CHR$(HI);
140 FORX=623 TO 637:READA:PRINT#5,CHR$(A);:NEXT
150 PRINT#5,CHR$(LF);:REM LEN NOME FILE
160 FORX=639TO657:READA:PRINT#5,CHR$(A);:NEXT
170 FORX=1TOLF:PRINT#5,MID$(NF$,X,1);:NEXT:REM INSTALLA NOME FILE
180 IFLF<16THENFORX=1TO16-LF:PRINT#5,CHR$(32);:NEXT
190 FORX=674TO769:PRINT#5,CHR$(PEEK(X));:NEXT
200 PRINT#5,CHR$(99);:PRINT#5,CHR$(2):REM VETTORE BASIC START DIROTTATO A $0263
210 CLOSE5:SCNCLR:SLOW:PRINT"F A T T O":END
220 DATA 169,000,032,144,255,032,066,193
230 DATA 169,141,002,003,142,003,003,169
240 DATA 000,162,008,168,032,186,255,169
250 DATA 162,146,160,002,032,189,255,169
260 DATA 000,166,045,164,046,032,213,255
270 DATA 076,166,090
280 END
```

```
DISASSEMBLATO RUN MAKER TRAMITE LOADER
              (listato 1)
-------------------------------------------
0263  LDA #$00    Disabilita i messaggi
0265  JSR $FF90   di sistema.
-------------------------------------------
0268  JSR $C142   Pulisce lo schermo.
-------------------------------------------
026B  LDA #$C6    Resetta i puntatori
026D  LDX #$4D    del BASIC START (o da
026F  STA $0302   programma li dirotta
0272  STX $0303   alla routine RUN).
-------------------------------------------
0275  LDA #$00    Apre un canale sulla
0277  LDX #$08    periferica 8 usando
0279  TAY         zero come indirizzo
027A  JSR $FFBA   secondario.
-------------------------------------------
027D  LDA #$ LF   Setta lunghezza (LF)
027F  LDX #$92    e posizione del nome
0281  LDY #$02    del file da caricare.
0283  JSR $FFBD
-------------------------------------------
0286  LDA #$00    Esegue un LOAD dalla
0288  LDX $2D     posizione inserita
028A  LDY $2E     nei puntatori di
028C  JSR $FFD5   inizio basic.
-------------------------------------------
028F  JMP $5AA6   Va a routine di RUN.
-------------------------------------------
0292 - 02A1       Nome file da caricare.
-------------------------------------------
```

```
5 REM C/128 AUTORUN MAKER
6 REM BY DOMENICO PAVONE
7 REM LISTATO N.2
8 :
10 COLOR0,1:COLOR4,1:COLOR6,1:COLOR5,16
20 SCNCLR:INPUT"FILE DA TRATTARE";NF$
30 IFLEN(NF$)>16ORLEN(NF$)<1THEN20
40 SCNCLR:INPUT"NOME FILE TRATTATO";RF$:RF=LEN(RF$):IFRF<1ORRF>16THEN40
50 SCNCLR:PRINT"INIBIZIONE DELLO STOP (S/N)?"
60 GETKEYX$
70 IFX$="S"THENLI=166:HI=90:REM PUNTATORI A 'RUN'
80 IFX$="N"THENLI=0:HI=11:REM PUNTATORI A $0B00
90 IFX$<>"S"ANDX$<>"N"THEN60
100 SCNCLR:FORX=606 TO 620:READA:POKEX,A:NEXT
110 POKE621,LI:POKE622,162:POKE623,HI:REM INSTALLA PUNTATORI
120 FORX=624 TO 637:READA:POKEX,A:NEXT
130 POKE638,RF:REM LEN NOME FILE TRATTATO
140 FORX=639 TO 657:READA:POKEX,A:NEXT
150 FORX=1TORF:POKE657+X,ASC(MID$(RF$,X,1)):NEXT
160 IFX$="N"THENFORX=2816 TO 2828:READA:POKEX,A:NEXT
170 PRINTCHR$(144);:POKE842,13:POKE843,13:POKE208,2:REM <CR> IN BUFFER TASTIERA
180 PRINT:PRINT:PRINT:PRINT"DLOAD";
190 PRINTCHR$(34)+NF$+CHR$(34):FORX=1TO4:PRINT:NEXT:PRINT"SYS606":PRINTCHR$(19)
200 NEW
210 REM ---- ROUTINE IN $25E -------
220 DATA 169,000,162,003,133,250,134,251
230 DATA 032,144,255,032,066,193,169
240 DATA 141,002,003,142,003,003
250 DATA 162,008,160,255,032,186,255,169
260 DATA 162,146,160,002,032,189,255
270 DATA 169,250,166,174,164,175,032,216
280 DATA 255,076,061,255
290 REM ---- ROUTINE IN $B00 -------
300 DATA 169,198,162,077,141,002,003
310 DATA 142,003,003,076,166,090
320 END
```

```
     DISASSEMBLATO AUTORUN MAKER
            (listato 2)
025E  LDA #$00    Prepara due puntatori
0260  LDX #$03    in pagina zero con
0262  STA $FA     indir.iniziale ($300)
0264  STX $FB     per la routine SAVE.
---------------------------------------
0266  JSR $FF90   Disabilita messaggi.
---------------------------------------
0269  JSR $C142   Pulisce lo schermo.
---------------------------------------
026C  LDA #$A6    Dirotta i puntatori di
026E  LDX #$5A    START BASIC verso la
0270  STA $0302   routine RUN ($5AA6) o
0273  STX $0303   altrove (da programma)
---------------------------------------
0276  LDX #$08    Apre un canale sulla
0278  LDY #$FF    periferica 8.
027A  JSR $FFBA
---------------------------------------
027D  LDA #$05    Setta lunghezza e
027F  LDX #$92    posizione del nome
0281  LDY #$02    del file oggetto.
0283  JSR $FFBD
---------------------------------------
0286  LDA #$FA    Salva la memoria da
0288  LDX $AE     $0300 fino alla
028A  LDY $AF     locazione puntata da
028C  JSR $FFD8   $AE e $AF (fine prog.)
---------------------------------------
028F  JMP $FF3D   Va a routine RESET.
---------------------------------------
0292 - 02A1       Nome file oggetto.
---------------------------------------
0B00  LDA #$C6    Resetta ai valori di
0B02  LDX #$4D    default i puntatori
0B04  STA $0302   di BASIC START.
0B07  STA $0303
---------------------------------------
0B0A  JMP $5AA6   Salta a RUN.
---------------------------------------
```

Test di coppia

Un programma, presente su Directory di questo mese, di notevole interesse per ricerche sul rapporto di coppia

di Claudio Romeo

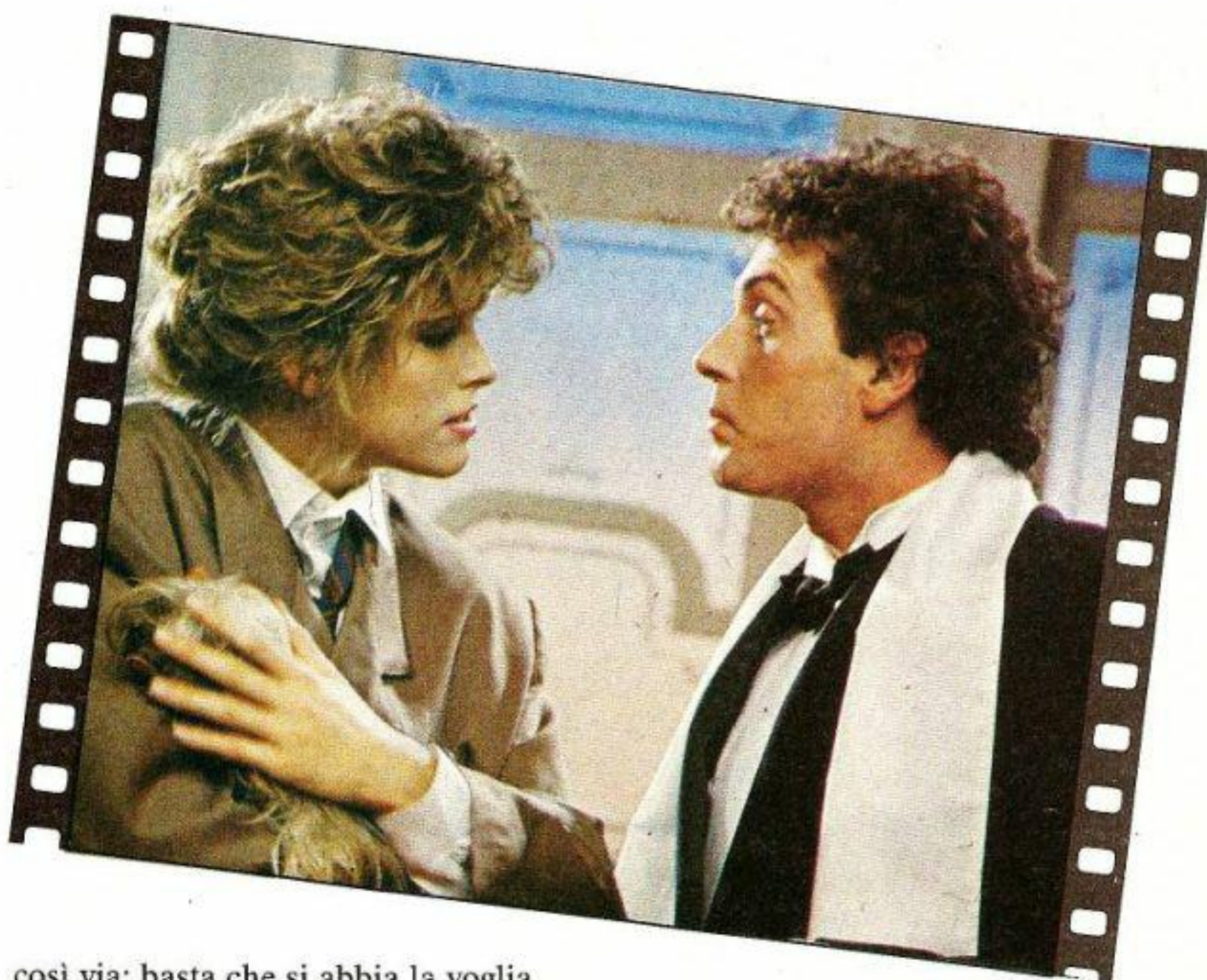
Lavorando nel campo sociale, spesso mi capita di constatare quanto poco si conoscano tra loro persone che, per legami di parentela, affettivi o di lavoro, passano anche molto tempo insieme.

Nel lavoro che svolgo quotidianamente individuo una delle conseguenze più drammatiche di una rete di relazioni non chiare; ma anche nella "normale" vita di tutti i giorni spesso si vivono tensioni che non esisterebbero se le persone che vivono insieme si conoscessero più nel profondo.

Il test presente su Directory è un programma (troppo lungo per trovare spazio su queste pagine) che, pur essendo un gioco, ha una certa credibilità scientifica (non per vantarmi, ma sappiate che ho meritato trenta all'esame di sociologia sostenuto presso l'Università Cattolica di Milano, con seminario sulla costruzione dei questionari d'indagine).

La validità del test non risiede tanto nel valore assoluto del punteggio finale, quanto nel rigore con cui le domande vengono formulate.

Che cosa fa

Vediamo che cosa è possibile stabilire con il test: il suo fine è quello di determinare la misura in cui i due componenti di una coppia si conoscano reciprocamente. La coppia può essere formata da marito e moglie, padre e figlio, amico e amico e così via: basta che si abbia la voglia di conoscersi.

Il programma pone una serie di domande (ne sono presenti 17 nella versione memorizzata su Directory), ognuna corredata da un certo numero di risposte "graduate"; ciò significa che queste sono all'interno di una "scala" ai cui estremi sono presenti due diversi modi d'approccio alla domanda stessa. E' quindi importante sottolineare che non esiste una risposta "giusta" sulla quale si fonda l'elaborazione del test, ma importante è determinare il differente modo con cui vengono "affrontate" le domande poste.

Inutile dire che è necessario riflettere con la massima attenzione prima di fornire una risposta, che deve essere la più onesta possibile.

Il gioco-test si deve svolgere tra due partecipanti alla volta ("A" e "B"). Mentre il primo, dopo il Run, risponde alle domande, l'altro deve spostarsi in un'altra stanza, in modo da evitare qualsiasi interferenza. Solo quando verrà richiesto dal programma, i due partecipanti si scambieranno il posto.

Al partecipante "A" il programma chiede, per una stessa domanda, due risposte: quella che, secondo "A", è

quella "corretta" e quella che suppone che "B" darà in seguito, quando toccherà il suo turno.

Se la previsione riguardo alla risposta che darà "B" risulterà corretta, allora vuol dire che "A" conosce bene "B".

Chiaro, no?

Il test misura inizialmente la quantità di "non conoscenza" che "A" ha di "B" basandosi sulla lontananza della previsione dall'effettiva risposta di "B" (le risposte sono graduate, ricordate?). Questa è indicata nel programma come variabile RO (Risultato Ottenuto). Tale valore assoluto viene quindi trasformato in percentuale, ed infine da percentuale di "non conoscenza" viene trasformato in percentuale di "conoscenza" (variabile PER).

Questo è, finalmente, il valore che verrà visualizzato alla fine delle vostre fatiche.

Disponendo della stampante sarà possibile selezionare l'apposita opzione che, oltre a mostrare i risultati, riporta tutte le domande corredate delle risposte fornite da entrambi i partecipanti in modo che vi possa essere uno scambio di idee ed un confronto: è proprio questo lo scopo ed il pregio del programma.

Un breve commento

Come già ricordato, il programma è troppo lungo (63 blocchi = 16 K circa) per esser pubblicato sulla rivista ed è questo l'unico motivo per cui è riportato solo su Directory, il dischetto che mensilmente la Systems editoriale offre per corrispondenza.

Va tuttavia ricordato che, da un punto di vista della programmazione, sono possibili ottimizzazioni notevoli, soprattutto se si decide di ricorrere alle semplificazioni proposte nell'inserto contenuto nel N.40 di C.C.C.

Il pregio del programma, comunque, è quello di dimostrare, ancora una volta, la notevole importanza che l'informatica può rivestire, anche nel quotidiano.

L'autore del programma svolge, infatti, il ruolo di operatore sociale in una comunità per il recupero dei tossicodipendenti, attività certamente delicata e difficile per chiunque.

E' con vero piacere, quindi, che dedichiamo spazio alla recensione sul lavoro svolto, ricordando che sarebbe opportuno, come sottolineato dallo stesso autore, uno scambio di esperienze e di programmi idonei tra coloro che, da vicino, vivono esperienze analoghe.

A.d.S.

Sotto il segno del trinomio

Un'applicazione matematica di indubbia utilità per gli studenti delle scuole superiori

di Armando Sforzi

Il trinomio di secondo grado è, notoriamente, un'espressione algebrica costituita da tre monomi ordinati, nella sua forma generale (detta anche canonica) secondo le potenze decrescenti di una variabile.

Di solito la forma tipica con la quale si presenta un trinomio è la seguente:

ax ↑2+bx+c

In taluni casi la forma può essere incompleta, essendo la costante "c" (e/o "b" nulla), ma non può mai risultare $a=0$ altrimenti non si parla più di trinomio di secondo grado.

Le lettere a, b, c che compaiono nell'espressione, denotano numeri reali o espressioni algebriche di valore determinato e noto, mentre la lettera x rappresenta la variabile. Questa può, di volta in volta, assumere diversi valori in corrispondenza dei quali il trinomio "reagisce" diventando positivo, negativo o annullandosi. In quest'ultimo caso i valori dell'incognita costituiscono le soluzioni o radici dell'equazione che si ottiene egualiando a zero il trinomio.

Dette radici sono tante quante ne indica il grado del polinomio (nel nostro caso due), anche se non è detto che siano sempre reali e distinte.

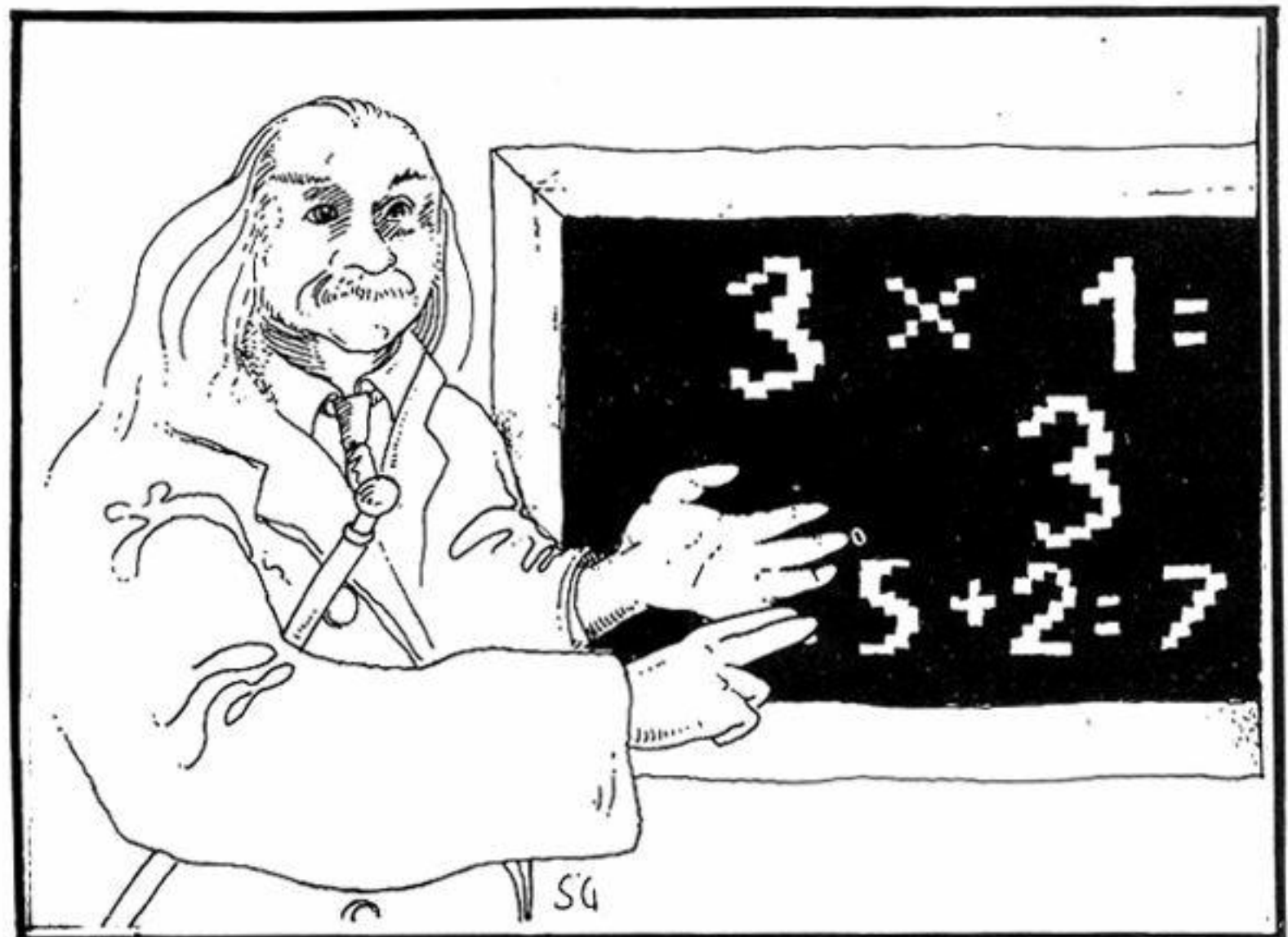
Se quindi volessimo studiare l'andamento del segno del nostro trinomio, la prima cosa che potrebbe venirci in mente di fare sarebbe quella di attribuire alla variabile x un numero abbastanza ampio di valori e registrare ogni volta il risultato ottenuto con lo scopo di disegnare un grafico del comportamento del segno.

Tutto questo lavoro, in effetti, può essere evitato ricorrendo al calcolo del discriminante del trinomio stesso (delta: b ↑ 2-4ac).

Infatti, conosciuto Delta, si possono presentare tre casi:

Delta maggiore di zero.

Le radici sono reali e distinte. Si può dimostrare che il trinomio assume segno uguale a quello del suo primo coefficiente "a", per valori esterni all'intervallo delle radici, cioè per x minore di x1 e per x maggiore di x2 (se, ovviamente, x1 risulta minore di x2).

Viceversa il segno del trinomio risulta opposto a quello del coefficiente "a", per valori compresi nell'intervallo delle due radici. Ricordiamo che per radice si intende quel valore di x secondo cui il trinomio si annulla.

Delta eguale a zero.

Le radici sono reali ma coincidenti: x1 = x2. Il trinomio, in questo caso, assume sempre il segno del suo primo coefficiente tranne che in corrispondenza di x1, per cui esso si annulla.

Delta minore di zero.

Le radici non esistono più nel campo reale. Il trinomio assume sempre il segno di "a" e non si annulla mai.

Con le regolette appena esaminate è abbastanza agevole studiare il segno di un trinomio, discutere disequazioni e sistemi di disequazioni, pur se con la limitazione del secondo grado.

Tuttavia il computer può renderci la vita più facile, in quanto può essere lui stesso che, opportunamente programmato, compie l'analisi e, magari, la visualizzazione semigrafica del risultato. Questo è proprio lo scopo per cui è stato scritto il programma che presentiamo.

Il programma

Dalla linea 110 alla linea 180 si accettano i valori dei coefficienti dei trinomi da studiare, si calcolano tutti i Delta e le radici.

Dalla linea 200 alla linea 290 le radici vengono ordinate per poterle stampare correttamente sul grafico.

Infine dalla linea 310 alla linea 470 è contenuta la routine di stampa del grafico. Alla linea 310 si fanno frequenti richiami ad un sottoprogrammma che è compreso tra le linee 540 e 610.

Tale subroutine è necessaria per trasformare una eventuale radice decimale in un numero frazionario. Infatti, osservando gli eserciziari di matematica, noteremo che invece di indicare le soluzioni in forma decimale, essi ricorrono alle notazioni frazionarie, radicali o esponenziali dei numeri, oltre che a quelle intere, è ciò per rendere più chiara e trattabile la quantità numerica stessa.

Digitato attentamente il listato, e dato il RUN, si indicherà il numero dei trinomi da analizzare (più di uno nel caso, ad esempio, di sistemi di disequazioni). Successivamente digiteremo i coefficienti dei trinomi nell'ordine richiesto (a, b, c).

Se il trinomio non si presentasse nella sua forma completa, se mancano, cioè, il coefficiente "b" o la costante "c", è necessario inserire uno 0 al loro posto.

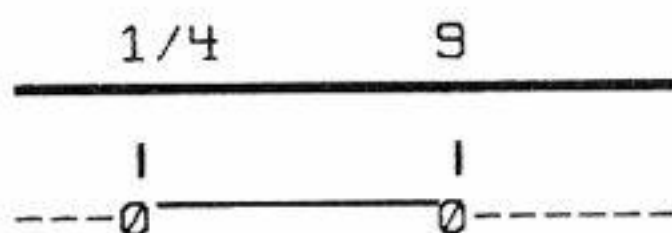
Atteso il tempo necessario per l'elaborazione dei dati (il BASIC è un linguaggio di bassa cilindrata quanto a velocità), sullo schermo apparirà il grafico dell'andamento dei segni, pronto per essere valutato.

Gli esempi

A questo punto è bene prendere in considerazione alcuni esempi. Sia dato il trinomio i cui coefficienti a, b, c valgono, rispettivamente:

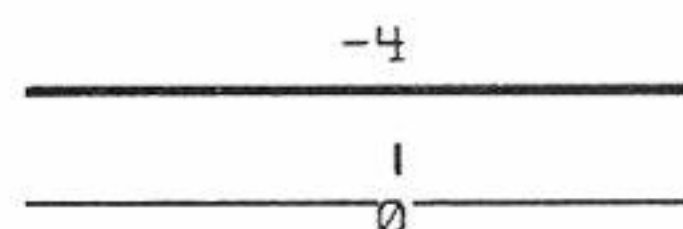
-4, 37, -9

Se digitiamo tali valori, alla richiesta del computer, otterremo il grafico di figura, da cui risulta che il trinomio in questione assume segno negativo (linea tratteggiata) per valori di x inferiori a 1/4, in corrispondenza del quale l'espressione si annulla (ecco la ragione della presenza dello zero).

```
Esempio 1:
(y=-4x↑2+37x-9)
```

Dunque la prima radice del trinomio vale 1/4. L'altra è 9 e tra questi due valori il trinomio mantiene il segno positivo (linea continua). Il segno tornerà ad essere negativo per valori di x superiori a 9. Come si vede, il computer mette a disposizione un comodo mezzo per conoscere il segno, le radici e, indirettamente, il Delta del trinomio, che in questo caso è positivo perchè le radici appaiono reali e distinte.

Esempio 2

Nel trinomio qui analizzato, i valori di a, b, c sono:
2, 16, 32
il grafico è quello di figura:

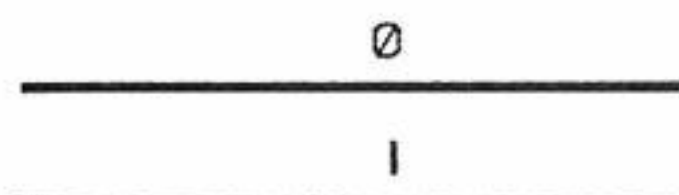
```
Esempio 2:
(y=2x↑2+16x+32)
```

Stavolta il grafico che appare informa che il nostro trinomio mantiene sempre un segno positivo per qualunque valore assegnato ad x, tranne che per x=4, per il quale si ha l'annullamento.

Ci troviamo, dunque, di fronte ad un'espressione in cui Delta è uguale a zero e le radici sono reali e coincidenti (valgono, cioè, entrambe -4).

Esempio 3

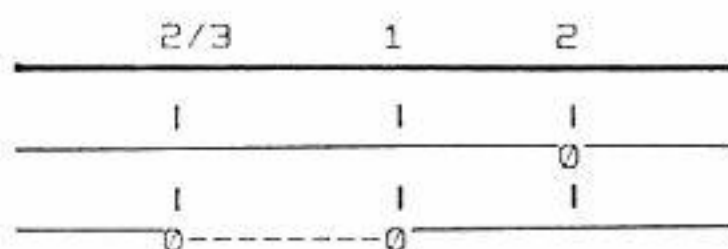
Consideriamo adesso una nuova espressione algebrica in cui valori di a, b, c, sono:
5, 8, 4.

Inserendoli nel computer otterremo il grafico di figura:

```
Esempio 3:
(y=5x↑2+8x+4)
```

Questo grafico, diverso dai precedenti, indica che il trinomio in qestione è sempre positivo, per qualunque valore di x e non si annulla mai (infatti manca lo 0 sulla linea del segno). Quello zero posto in alto al centro, sulla linea delle radici, ha un valore puramente simbolico e serve, cioè, solo a mostrare che non esistono valori negativi o positivi da attribuire all'incognita tali che annullino il trinomio. E' questo il caso del Delta negativo.

Esempio 4
Analizziamo, ora, la disequazione frazionaria della figura che segue:

```
Esempio 4:

x↑2-4x+4
────────── >0
3x↑2-5x+2
```

Nonostante l'apparenza, questa frazione è semplicemente formata da due trinomi di secondo grado i quali, per soddisfare la disequazione (e fare in modo che tutta l'espressione sia positiva) devono assumere entrambi segno uguale (entrambi positivi o entrambi negativi).
Vediamo, dunque, in quale ambito la disequazione sarà soddisfatta.
(1) a, b, c : 1, -4, 4
(2) a, b, c : 3, -5, 2
Il grafico che apparirà sullo schermo sarà quello appena esaminato in figura.

Osservandolo bene si evince che la disequazione sarà soddisfatta per x minore di 2/3 e per x maggiore di 1 (entrambe le linee positive), con x diverso da 2 perchè in quel punto, annullandosi il trinomio numeratore, la frazione algebrica varrà zero.

Esempio 5
Consideriamo, in ultimo, un sistema di disequazioni come quello che segue:

```
Esempio 5:

| x↑2- 4x >0
| 2x↑2 +5x -7 >0
| x↑2 -6x >0
```

I coefficienti da digitare, questa volta, saranno i seguenti:
(1) a, b, c : 1, -4, 0
(2) a, b, c : 2, 5, -7
(3) a, b, c : 1, -6, 0
Il sistema considerato, avendo posto tutte le sue disequazioni positive, sarà soddisfatto per valori di x minori di -7/2 e maggiori di 6, perchè solo il quelle zone i tre trinomi assumono contemporaneamente il segno positivo.

Il lettore, a questo punto, dovrebbe essere in grado di utilizzare il programma risolvendo i numerosi esercizi riportati sui libri di testo di matematica.

SCHEDA TECNICA

Software didattico per applicazioni matematiche.

Idoneo per computer C/64, C/16, Plus/4, C/128, Vic-20.

Consigliato agli studenti delle scuole superiori.

Anche il programma pubblicato in queste pagine è contenuto nel disco "Directory" di questo mese.

```
10 REM   * SEGNO DEL TRINOMIO *
20 REM   * PER QUALSIASI COMMOD
   ORE
30 REM   * BY ARMANDO SFORZI 19
   87
40 REM   * BOZZANO (LU)
50 :
90 DIM A,B,C,J,JJ,K,NR,NT,A$
100 PRINT"[CLEAR] * [RVS]  SEGN
    O DEL TRINOMIO DI 2' GRADO[
    RVOFF] *":PRINT
110 INPUT "NUMERO TRINOMI";NT:D
    IM T(NT,4),S$(4),R(10)
120 S$(0)=CHR$(96):S$(2)=CHR$(4
    5):S$(4)=CHR$(96)
130 PRINT"COEFFICIENTI:":FOR J=
    1 TO NT:PRINT"("J")  ";
140 INPUT "A,B,C";A,B,C:IF A=0
    THEN PRINT"ERRORE":GOTO 140
150 T(J,1)=B*B-4*A*C:T(J,2)=A:I
    F T(J,1)<0 THEN 180
160 T(J,3)=(-B+SQR(T(J,1)))/(2*
    A):T(J,4)=(-B-SQR(T(J,1)))/
    (2*A)
170 IF T(J,3)>T(J,4) THEN A=T(J
    ,3):T(J,3)=T(J,4):T(J,4)=A
180 NEXT
190 REM   * ORDINA LE RADICI *
200 FOR J=1 TO NT:IF T(J,1)<0 T
    HEN 230
210 NR=NR+1:R(NR)=T(J,3):IF T(J
    ,1)=0 THEN 230
220 NR=NR+1:R(NR)=T(J,4)
230 NEXT
```

```
240 FOR J=1 TO NR-1:A=R(J):FOR
    JJ=J TO NR:IF A>R(JJ) THEN
    B=A:A=R(JJ):R(JJ)=B
250 NEXT:R(J)=A:NEXT:IF NR=1 TH
    EN 310
260 REM  * ELIMINA LE RADICI UG
    UALI *
270 A=1:FOR J=2 TO NR:IF R(J)=R
    (J-1) THEN 290
280 A=A+1:R(A)=R(J)
290 NEXT:NR=A
300 REM  * RAPPRESENTAZIONE GRA
    FICA *
310 PRINT"[CLEAR][3 DOWN]":C=IN
    T(35/(NR+1)):FOR K=1 TO NR:
    A=R(K):GOSUB 540
320 PRINT TAB(C*K-1)A$;:NEXT:PR
    INT
330 PRINT TAB(1);:FOR J=1 TO 38
    :PRINTCHR$(162);:NEXT:PRINT
```

```
340 FOR J=1 TO NT*2:FOR K=1 TO
    NR:PRINT TAB(C*K)CHR$(98);:
    NEXT:PRINT:NEXT
350 PRINT"[HOME][6 DOWN]"
360 FOR K=1 TO NT:PRINT TAB(1);
370 IF T(K,1)<0 THEN FOR J=1 TO
     38:PRINTS$(1- SGN(T(K,2)))
    ;:NEXT:GOTO 470
380 IF T(K,1)>0 THEN 420
390 A=1:B=1
400 GOSUB 520:IF R(A)<>T(K,3) T
    HEN A=A+1:GOTO 400
410 PRINTCHR$(157)"0";:FOR J=PO
    S(0) TO 38:PRINTS$(1- SGN(T
    (K,2)));:NEXT:GOTO 470
420 A=1:B=1
430 GOSUB 520:IF R(A)<>T(K,3) T
    HEN A=A+1:GOTO 430
440 PRINTCHR$(157)"0";:A=A+1:B=
    3
450 GOSUB 520:IF R(A)<>T(K,4) T
    HEN A=A+1:GOTO 450
460 PRINTCHR$(157)"0";:FOR J=PO
    S(0) TO 38:PRINTS$(1- SGN(T
    (K,2)));:NEXT
470 PRINT:PRINT:NEXTK:PRINT"[HO
    ME]1.[RVS]ANCORA[RVOFF]  2.
    [RVS]FINE[RVOFF]"
480 GET A$:IF A$="" THEN 480
490 IF A$="1" THEN CLR :GOTO 90
500 IF A$="2" THEN PRINT"[CLEAR
    ]":END
510 GOTO 480
520 FOR J=1 TO C:PRINTS$(B- SGN
    (T(K,2)));:NEXT:RETURN
530.REM  * TRASFORMA NUMERO DEC
    IMALE IN FRAZIONARIO *
540 A$="":B=0:IF A<0 THEN B=1
550 FOR J=1 TO 15:FOR JJ=2 TO 1
    5:IF ABS(J/JJ-ABS(A))<.01 T
    HEN 580
560.NEXTJJ,J:IF A$="" THEN A$=M
    ID$(STR$(A),1,5)
570 RETURN
580 A$=STR$(J)+"/"+RIGHT$(STR$(
    JJ),1-(JJ>9))
590 IF J/JJ=INT(J/JJ) THEN A$=S
    TR$(J/JJ)
600 IF B THEN A$="-"+MID$(A$,2)
610 J=15:JJ=J:GOTO 560
```

PRODOTTI SYSTEMS EDITORIALE

Software su cassetta

La voce III	L.12000
Raffaello	L.10000
Oroscopo	L.12000
Computer-Music	L.12000
Gestione familiare	L.12000
Banca dati	L.12000
Dichiarazione dei redditi (740/S)	L.16000
Matematica finanziaria	L.20000
Analisi di bilancio	L.20000
Arredare (richiede linguaggio Simon's Basic)	L.10000

Software su disco

Ms-Dos & Gw-Basic	L.25000
Ms-Dos & Gw-Basic (con prova di acquisto cassetta)	L.15000
La voce III	L.12000
Raffaello	L.10000
Oroscopo	L.12000
Computer-Music	L.12000
Gestione familiare	L.12000
Banca dati	L.12000
Dichiarazione dei redditi (740/S)	L.24000
Matematica finanziaria	L.20000
Analisi di bilancio	L.20000
Arredare (richiede linguaggio Simon's Basic)	L.20000
Graphic Expander C/128 in modo 80 colonne	L.27000
Linguaggio macchina + Routine grafiche	L.12000
Offerta speciale fascicoli "Commodore" 13, 14, 15 e "Personal Computer" N.1, 2, 3	L.25000
Offerta speciale fascicoli "Personal Computer" N.2, 3, 5	L.12000
Offerta speciale fascicoli "Commodore" 13, 14, 15 e "Personal Computer" N.1, 2, 3, 5	L.27000
Offerta "Commodore speciale L.M." + dischetto	L.16000

Libri

64 programmi per il Commodore 64	L. 4800
I miei amici C/16 e Plus/4	L. 7000
Strategie vincenti per Commodore 64	L. 5800
62 programmi per Vic 20, C/16 e Plus/4	L. 6500
Utilities e giochi didattici	L. 6500
Tutti i segreti dello Spectrum	L. 7000
Simulazioni e test per la didattica	L. 7000
Impara giocando il Basic dello Spectrum	L. 7000
Micro Pascal per Commodore 64/128	L. 7000
Dal registratore al drive del C/64	L. 7000
Ada	L. 5000
Il linguaggio Pascal	L. 5000

Directory

Ciascun dischetto	L.12000

Arretrati

Ciascun numero arretrato di Commodore Computer Club	L. 5000
Ciascun numero arretrato di Personal Computer	L. 5000
Ciascun numero arretrato di VR Videoregistrare	L. 5000

Per un ottimale utilizzo del software "Matematica finanziaria" è opportuna la lettura degli articoli relativi pubblicati sui N.13, 14, 15 della rivista "Commodore" e sui N.1, 2 e 3 della Rivista Personal Computer.

Per un ottimale utilizzo del software "Analisi di Bilancio" è opportuna la lettura degli articoli relativi pubblicati sui N.2, 3, 5 della Rivista Personal Computer.

Per un ottimale utilizzo del software "Linguaggio Macchina e Routine grafiche per C/64" è opportuna la lettura del fascicolo "Commodore Speciale" appositamente dedicato.

Coloro che desiderano procurarsi i prodotti della Systems Editoriale devono inviare, oltre alla cifra risultante dalla somma dei singoli prodotti, la cifra di L.3000 per spese di imballo e spedizione, oppure L.6000 se si preferisce la spedizione per mezzo raccomandata.

Sconti e agevolazioni

Le spese di imballo e spedizione sono a carico della Systems se ciascun ordine è pari ad almeno L.50000 (di listino).

Gli abbonati hanno diritto allo sconto del 10% e alla spedizione gratuita se la somma totale raggiunge la cifra di L.50000 (di listino).

Oltre alla spedizione gratuita, viene praticato uno sconto del 10% (per gli abbonati è del 20%) se la cifra raggiunta per ciascun ordine raggiunge le L.100000 (di listino).

Abbonamenti

Commodore Computer Club (11 fascicoli)	L.40000
Personal Computer (11 fascicoli)	L.40000
Commodore Computer Club + Personal Computer (11 + 11 fascicoli)	L.65000
VR Videoregistrare (12 numeri)	L.45000

N.B.: la cifra per gli abbonamenti non può essere conteggiata per ottenere gli sconti e le agevolazioni di cui sopra.

Non è assolutamente possibile inviare materiale contrassegno.

Compilate un normale modulo di C/C postale indirizzando a:

C/C postale N. 37952207
Systems Editoriale
Viale Famagosta, 75
20142 Milano

Non dimenticate di indicare chiaramente, sul retro del modulo (nello spazio indicato con "Causale del versamento") non solo il vostro nominativo completo di recapito telefonico, ma anche il materiale desiderato.

In ogni caso sarebbe opportuno inviare la presente scheda, debitamente compilata, allegando la fotocopia della ricevuta del versamento effettuato.

Chi volesse ricevere più celermente la confezione deve inviare la somma richiesta mediante assegno circolare oppure normale assegno bancario (non trasferibile o barrato due volte) intestato a: Systems Editoriale - Milano.

Quale fascicolo manca alla tua enciclopedia Commodore?

Per ordinare i fascicoli mancanti alla tua collezione di Commodore Computer Club utilizza l'apposita scheda in fondo alla rivista.

Principi di multi programmazione

Time sharing, multi-tasking, interrupt: tanti nomi per indicare un concetto che, a pensarci bene, conosciamo benissimo.

di Alessandro de Simone

Ab ovo

Molto spesso, iniziando a studiare un nuovo argomento, si è convinti di affrontare temi nuovi, difficili e, in ogni caso, impegnativi.

Si dimentica, invece, che qualsiasi trattazione deve necessariamente avere una base logica, semplice e sicuramente comprensibile, proprio perchè è compiuta dall'uomo.

Provate a considerare una qualsiasi invenzione, come ad esempio l'automobile: i sedili hanno forma anatomica (i marziani, forse, li fabbricherebbero diversamente), il tetto è presente per ripararci dalle intemperie, il parabrezza dal vento, le serrature dagli... indesiderati.

Anche il calcolatore, naturalmente, non sfugge alla regola elementare che gli impone di avere forma e funzioni fatte "a misura d'uomo": la tastiera ha senso proprio perchè sappiamo muovere le dita con una certa rapidità e disinvoltura (un cavallo avrebbe certamente inventato qualcosa di diverso), lo schermo richiama la struttura di un qualsiasi testo scritto, i colori del video si rendono necessari per evitare all'uomo, sempre lui, di ritrovarsi in un "ambiente" privo di variazioni cromatiche con le quali è in costante contatto.

E così, di seguito, possiamo pensare al telefono, all'ascensore, all'aereo, agli alberghi e ad un qualsiasi altro oggetto (o servizio) inventato dall'uomo.

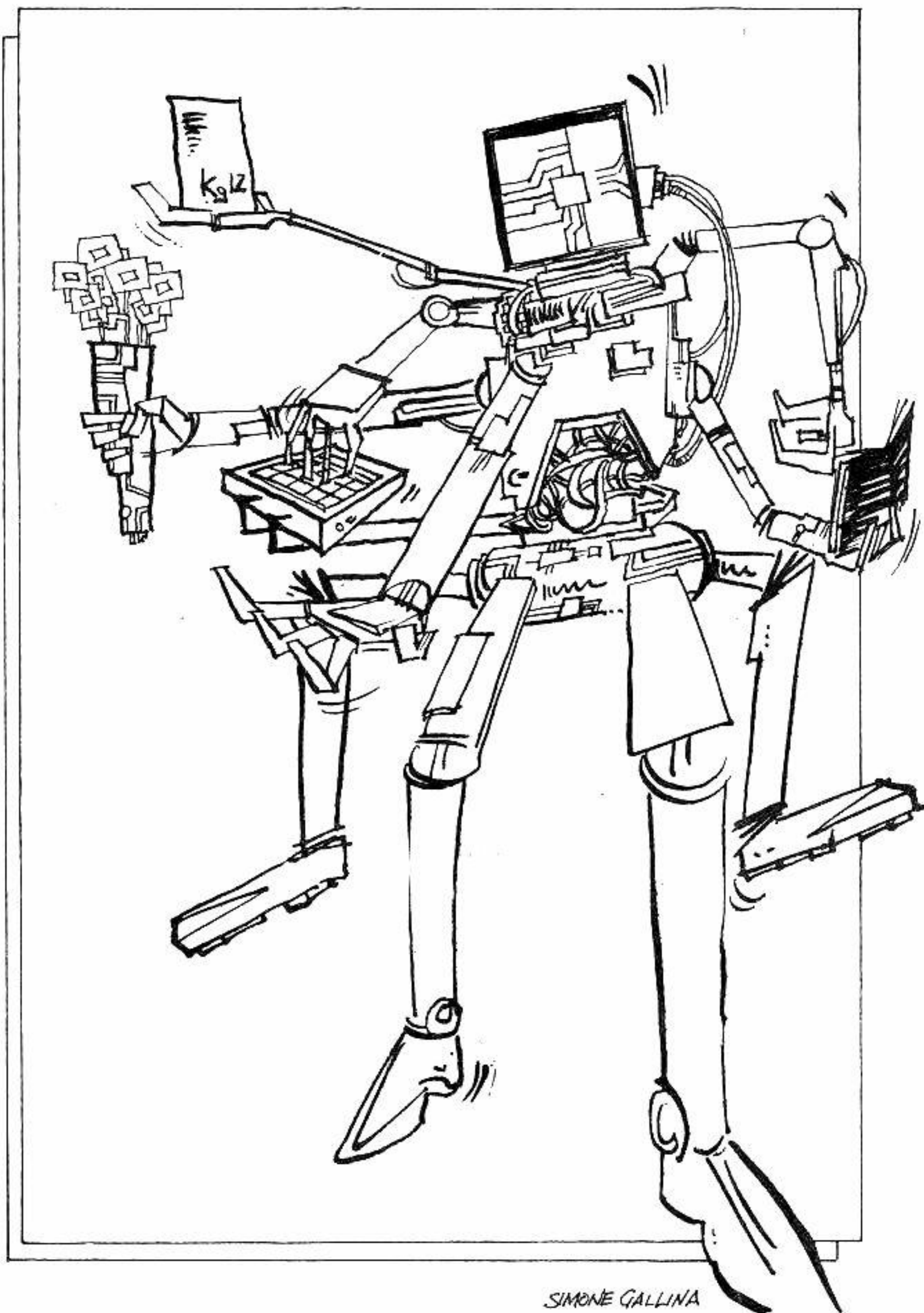
Ma se finora abbiamo considerato gli oggetti (hardware) non possiamo trascurare le idee (software) che, anch'esse, non sfuggono alla complessa struttura mentale dell'uomo che le ha generate.

L'uomo

Vi diamo cinque secondi di tempo per dire che cosa state facendo in questo momento; è probabile che vi limitiate a rispondere "sto leggendo" ritenendo, in perfetta buona fede, di aver esaurito l'argomento, con tale semplice risposta.

Vediamo in dettaglio, al contrario, che cosa state facendo realmente:

- Le vostre mani stanno reggendo il giornale con una angolazione tale da consentire alla luce disponibile di illuminare le pagine con la giusta intensità.

SIMONE GALLINA

- Gli occhi stanno rapidamente scorrendo i caratteri di ciascun rigo seguendo una procedura rigorosa (da sinistra a destra e dall'alto in basso).
- Il vostro corpo è posizionato in perfetto equilibrio (anche se siete ignobilmente stravaccati in poltrona) e tutti i muscoli del vostro corpo sono impegnati, chi più chi meno, a tenerlo in tale equilibrio.
- Il vostro cuore sta pulsando alla giusta velocità per distribuire il sangue lungo arterie, vene, venuzze e canalicoli vari.
- I vostri polmoni stanno respirando per immettere ossigeno nel corpo ed estrarne anidride carbonica.
- ...e potremmo smetterla qui, prima di esaminare le imbarazzanti operazioni in corso verso il sud del vostro corpo.

Il corpo umano, insomma, è una macchina incredibilmente potente e, fortuna per noi, costantemente attiva nell'espletamento di migliaia di operazioni da svolgere in contemporanea e, in moltissimi casi, in perfetto sincronismo.

E se, con un po' di fantasia, riesaminiamo l'automobile di prima con analogo spirito di osservazione, non possiamo fare a meno di rilevare che, in uno stesso istante, lo spinterogeno compie una determinata funzione, alcune valvole sono aperte, ed altre provvidenzialmente chiuse, un ammortizzatore evita di ripercuotere sui passeggeri gli effetti di un fosso, la cinghia della ventola fa in modo di raffreddare il liquido che, a sua volta, è spinto nel circuito da un'altra valvola mossa dall'albero motore che fa girare l'alberino dello spinterogeno... e così via.

Il concetto di più funzioni svolte contemporaneamente, quindi, non è affatto nuovo per noi; solo che, magari, non ci è mai capitato, finora, di rifletterci sufficientemente.

Dal cervello al micro

Il cervello umano, come è noto, è ancora piuttosto misterioso e le funzioni che svolge, addirittura, sono quasi tabù. Nessuno è in grado di spiegare certi fenomeni, atteggiamenti e comportamenti umani.

L'uomo è una macchina rigorosamente complessa, difficile da esplorare ed influenzabile, per giunta, da fattori che, a loro volta, possono variare il "percorso" abituale del flusso del pensiero.

Non staremo qui a ricordare il mistero della parapsicologia, dell'autosuggestione o dei fenomeni psicosomatici. Nè, tantomeno, tenteremo di spiegarci le presunte potenzialità degli astro-chiro-carto-manti che offrono le loro salatissime prestazioni solo per per appuntamento e al di fuori, ovviamente, di verifiche fiscali.

Vogliamo solo dire, invece, che una macchina complessa come il cervello umano può, alla luce delle moderne conoscenze, essere osservata "da lontano", con attenzione e umiltà, tentando di intuirne i modi di funzionamento.

Per spiegare uno stesso fenomeno umano o sociale, insomma, qualsiasi teoria può essere valida; altrettanto valida può essere, tuttavia, una teoria esattamente contraria. Ne è una riprova il proliferare, nel corso dei millenni, di numerose scuole filosofiche, credenze religiose, ideologie politiche ed ipotesi scientifiche che hanno tutte in comune, oltre al tentativo di escludersi reciprocamente, la ricerca della cosiddetta verità, finalizzata, o meno, a scopi che si ritengono di importanza escatologica.

Pur non potendo intrattenerci a lungo sulla Verità, che qui in Systems possediamo tutti, ci limiteremo a considerare gli aspetti che più interessano per meglio comprendere le applicazioni ai computer in generale, e Commodore in particolare.

L'impero dei sensi

Si dice che abbiamo a disposizione cinque sensi (vista, udito, tatto, olfatto e gusto) ma sono in molti ad affermare che ne disponiamo in maggior numero.

Se consideriamo l'udito, ad esempio, ci accorgiamo che questo è sempre attivo, in qualsiasi istante, anche durante il sonno; è questo, e non altri, il motivo per cui le sveglie, per funzionare, sono dotate di dispositivi sonori (bip-bip, radio, musicassette, eccetera). Se, nel tentativo di svegliare una persona, provate ad accendere e spegnere la luce della stanza, è probabile che non otteniate il risultato voluto; analogamente, se provate a diffondere un profumo, è più probabile che il suo effetto produca variazioni nel sogno in corso. Solo un pizzicotto ben assestato può avere l'effetto di una sveglia, ma le reazioni che si osservano nella maggior parte dei soggetti trattati in tal modo hanno finora consigliato i fabbricanti di sveglie ad utilizzare il tradizionale senso dell'udito per interrompere il sonno.

Un calcolatore dispone anch'esso dei sensi; magari non sono cinque nè raggiungono i livelli di sofisticazione propri dell'uomo, ma, nel loro piccolo, possono esser considerati sufficientemente "sensibili".

La tastiera è, per il computer, ciò che per noi è l'orecchio; i vari comandi da impartire, e le istruzioni che desideriamo siano eseguite, possono essere inviate tramite tastiera che, come è noto, dispone di un tasto vitale per la conferma: il tasto Return, che non a caso è, in genere, il tasto più grosso presente.

Un computer, però, dispone anche di altre orecchie: pensate al modem, dal quale può ricevere informazioni, oppure al drive, dal quale attinge dati; e considerate il nuovo strumento rivoluzionario, il mouse, che le orecchie sembra averle per davvero. Altre orecchie sono la tavoletta grafica, la penna ottica, il joystick e tutti quegli strumenti che, insomma, possono "comunicare" al calcolatore un ordine, un'istruzione, una scelta.

Ma la tecnologia moderna offrirà, tra brevissimo tempo, un vero "orecchio", e intendiamo, con ciò, riferirci al comune microfono. Tutti sanno che, anche oggi, è possibile digitalizzare un suono e, in seguito, riascoltarlo per mezzo del computer; ma in un prossimo futuro saranno presenti sul mercato computer in grado di riconoscere la voce dell'utilizzatore e, ovviamente, di "capire" gli ordini da espletare. E tutto ciò non avverrà fra chissà quanto tempo. Esistono di già calcolatori in grado di fare quanto descritto, ma si è ancora in fase sperimentale e le applicazioni, almeno per il momento, sono limitate. Uno dei problemi dei computer della quinta generazione, disponibili certamente entro la fine di questo secolo (che non è poi tanto lontana) è proprio la possibilità di comunicare a viva voce con una macchina.

Il video, con un po' di fantasia, può esser considerata la "voce" del computer. La visualizzazione di dati, di informazioni e di risultati di elaborazioni assolve certamente al compito di "dialogare" con l'utilizzatore che, appunto, riceve una risposta ai suoi quesiti tramite video.

In effetti la "voce" dei primi calcolatori era rappresentata dalla stampante che, nei decenni passati, era lo strumento più economico per fornire i risultati delle elaborazioni. In seguito, però, grazie ai progressi della tecnologia, fu possibile adattare un'altra invenzione dell'uomo, la TV, alla visualizzazione dei dati; è facile immaginare il risparmio di carta che seguì alla

diffusione di questa nuova "voce" dei calcolatori; al giorno d'oggi, infatti, viene riportato su carta solo ciò che è assolutamente indispensabile. In un futuro, che speriamo prossimo, la carta risulterà superflua come supporto, sostituita da dischi magnetici in grado di visualizzare su schermi le informazioni desiderate; parecchi alberi, quindi, potranno non esser destinati alle cartiere grazie alla minor richiesta del mercato. Se la burocrazia italiana potesse fare a meno di carte e cartacce, potremmo disporre di un maggior numero di foreste e, magari, di una maggiore efficienza in alcuni servizi. Quando, fra pochi(?) anni la posta cartacea sarà sostituita (come accade nei paesi veramente industrializzati) da quella elettronica, ne riparleremo.

La "voce" di un computer è rappresentata, oltre che da un video o da una stampante, anche dal plotter, in grado di disegnare con vere e proprie penne, dal modem e dal drive (che assumono, quindi, la duplice funzione di orecchia e bocca).

Recentemente una vera voce, digitalizzata opportunamente, è stata introdotta come strumento per risolvere il problema del colloquio. Chi possiede il C/64 sa benissimo che risulta straordinariamente semplice dotare il piccolo computer di un sintetizzatore vocale. Chi possiede l'Amiga, poi, può ricorrere frequentemente alla divertente, quanto fedelissima, riproduzione di frasi sintetizzate.

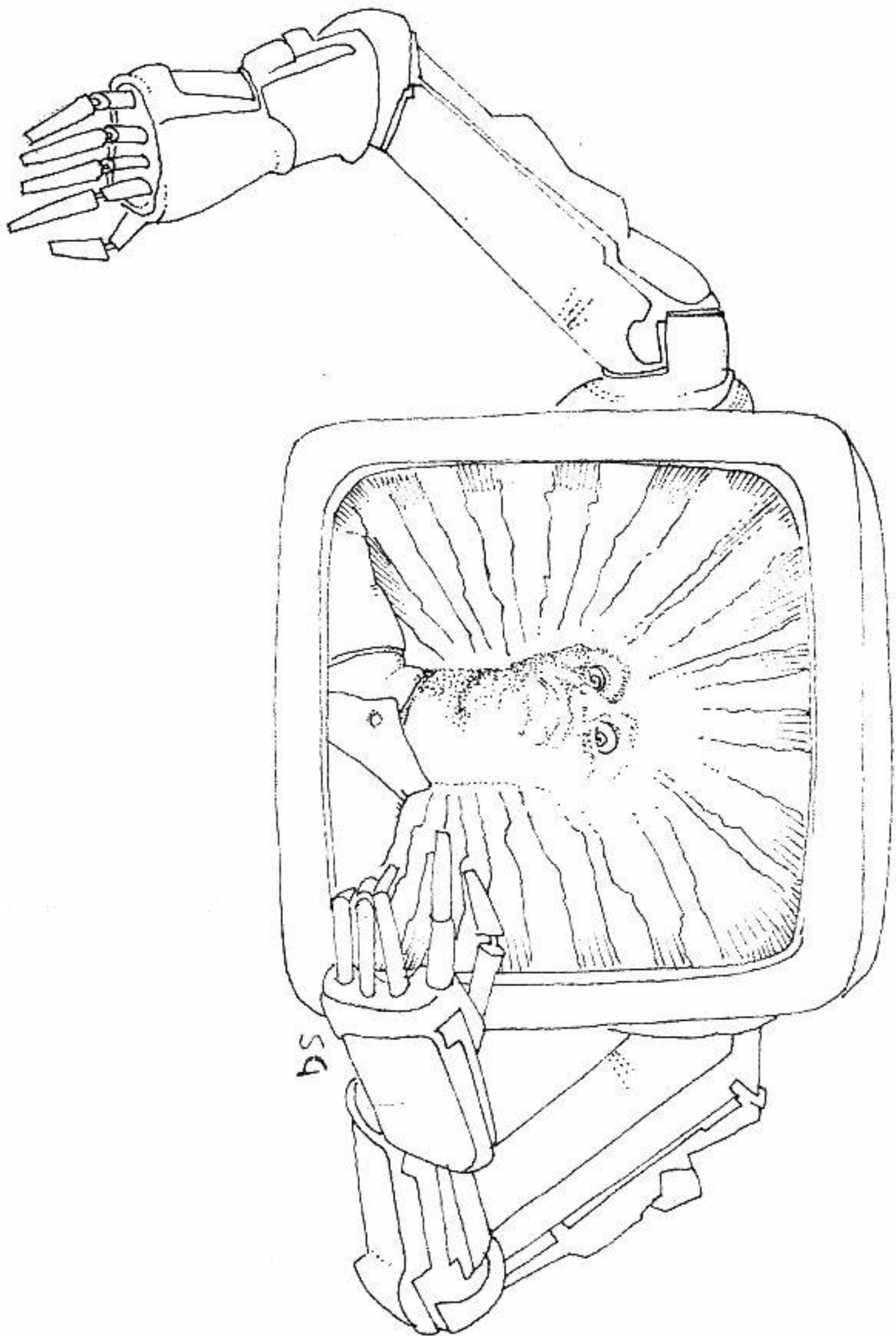
Per ciò che riguarda il gusto, a onor del vero, siamo ancora lontani dalle applicazioni pratiche, a meno che non vogliamo considerare i vari sensori oggi disponibili (analizzatori di sostenze gassose, chimiche, radioattive eccetera) come percettori di odori.

Per la simulazione del tatto, invece, non c'è che l'imbarazzo della scelta fra sensori elettromeccanici, sensori di pressione, di allungamenti ed altre diavolerie in grado di riconoscere la variazione, anche minima, di forma o configurazione di un oggetto.

Ma come fa, un computer, a controllare i suoi sensi?

I controlli

La "macchina" umana, per funzionare, ha bisogno di un direttore d'orchestra, il cervello, altamente affidabile e veloce. Consideriamo, ad esempio, che cosa accade quando compia-

mo un'operazione apparentemente banale, come il passeggiare lungo una strada.

In tale azione "dinamica" i controlli da effettuare sono incredibilmente numerosi e molti di essi, tra l'altro, devono essere eseguiti in tempo reale. Per mantenere il corpo eretto, ad un certo gruppo di muscoli è affidato il compito di entrare in trazione ed in compressione, molto rapidamente, affinchè il baricentro del corpo risulti posizionato, istante per istante, all'interno di una particolare zona che garantisca l'equilibrio. Non solo, poi, tale controllo deve essere effettuato con precisione, ma, soprattutto, eventuali correzioni di "assetto" devono essere compiute immediatamente (in tempo, quindi, reale), per evitare di cadere rovinosamente sul marciapiedi stesso.

In genere tale controllo, e conseguente modifica di assetto, avviene con sufficiente rapidità; altre volte, invece, i muscoli non rispondono prontamente agli stimoli, oppure gli stimoli che impongono il cambio di posizione non giungono in tempo: è questo il motivo per cui qualche volta cadiamo per un passo falso, oppure inciampiamo in qualche ostacolo che non avevamo notato; ed è anche ciò che accade, purtroppo, agli automobilisti che si ostinano a guidare in stato di ebbrezza, che impedisce la prontezza di riflessi o, se preferite, la possibilità di agire in tempo reale.

Naturalmente, nel corso della nostra passeggiata, non basta verificare costantemente la posizione del baricentro (che, come l'azione del respirare, è assolta in modo inconsapevole); bisogna effettuare altri controlli: considerare eventuali ostacoli ed evitarli (altra gente che cammina sullo stesso marciapiede, macchine parcheggiate selvaggiamente, ricordini di cani, eccetera); optare per eventuali variazioni di percorso (vetrine interessanti, incontri di conoscenti con cui fermarsi a chiacchierare, soste per riposarsi nel caso in cui trasportiamo pacchi pesanti, e così via).

A tutte queste funzioni, che vanno svolte in contemporanea, in sincronismo ed in tempo reale, provvede il cervello secondo una tecnica che, in parte, è ancora sconosciuta.

Analogamente un calcolatore deve esser strutturato in modo da espletare più compiti contemporaneamente, alcuni dei quali in sincronia con altri.

Pensate, infatti, al vostro calcolatore nel momento in cui lo accendete: sembra che non accada nulla e che non sia compiuta alcuna elaborazione; il cursore lampeggia, è vero, ma

difficilmente qualcuno può essere indotto a definire "elaborazione" quel semplice cambiamento di colore di un quadratino del video!

Ma esaminiamo con maggiore attenzione che cosa, realmente, sta accadendo in un qualsiasi istante durante il quale il computer sembra che si limiti a far lampeggiare il cursore:

Una particolare routine in linguaggio macchina sta esaminando se non sia stato premuto un tasto; sapete benissimo, infatti, che se premete un tasto questo viene immediatamente visualizzato sullo schermo e ciò dimostra, appunto, che c'era "qualcuno" che si aspettava la pressione di un tasto da un momento all'altro.

Ma mentre qualcuno (o qualcosa) controlla continuamente la tastiera, qualcun altro (o qualche altra cosa) deve controllare il pennello elettronico del video e imporre di illuminare un pixel, o meno, a seconda di come è richiesto in quel particolare momento; solo in questo modo, infatti, è possibile che un eventuale messaggio rimanga costantemente visualizzato sullo schermo. C'è, ovviamente, anche una routine che assegna un ben determinato colore ai pixel presenti sul video; e deve esser presente anche una routine che esamina l'eventuale variazione di stato relativo al connettore dei joystick. Stavamo per dimenticare la routine che aggiorna di continuo l'orologio interno e, nel caso del C/64, anche quella dell'orologio e della sveglia gestibili in modo particolare; per non parlare del controllo dello stato degli sprite (se ve ne sono) e, se è il caso, del chip sonoro.

E tutto ciò (insieme a molte altre cose ancora) avviene durante il semplice lampeggio del cursore; l'avreste mai immaginato?

Ma se il microprocessore è uno solo, come è possibile che tutte queste funzioni vengano eseguite contemporaneamente? Sappiamo benissimo, infatti, che se premiamo un tasto questo viene visualizzato immediatamente, magari durante la visualizzazione costante di altri caratteri (e sprite) eventualmente già presenti sullo schermo, mentre un nostro amico si diverte a maneggiare il joystick e, perchè no, mentre un suono continuo viene emesso grazie al precedente settaggio delle opportune poke del chip sonoro.

Come fa il microprocessore, dunque, a fare contemporaneamente tutte queste cose? E come potrà, a maggior ragione, controllare tutte le azioni ora viste quando gli verrà richiesto di far

girare un programma complesso, come un videogame in linguaggio macchina che richiede la gestione di grafica a colori in alta risoluzione, con controllo di due joy, con musica di sottofondo e seguendo una strategia di gioco rapida, costantemente variabile ed incredibilmente complessa?

La partizione del tempo

Sicuramente avete sentito parlare di produttività e il modo di esprimere le unità di misura adoperate possono, in certi casi, lasciare perplessi.

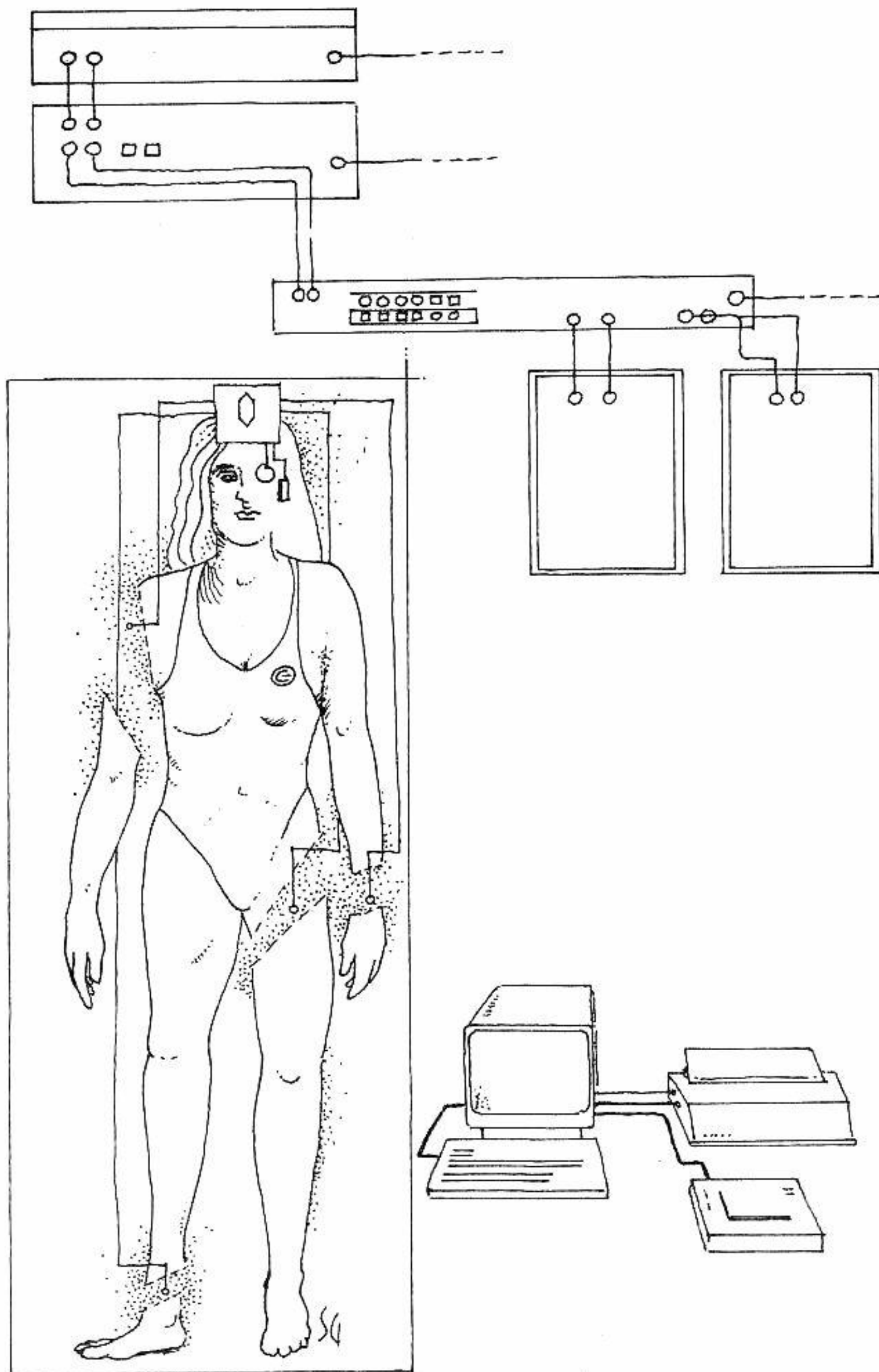
Considerando, ad esempio, la produttività dell'operaio giapponese addetto alla fabbricazione di automobili, si dice che questi, in media, produca 12 autovetture.

Ciò non significa che al singolo operaio è affidato il compito di realizzare, per intero, una vettura completa. Sappiamo benissimo che in Giappone, come in Italia e in qualsiasi fabbrica di auto, vi sono operai specializzati nel montare le parti elettriche, la carrozzeria, il motore, gli allestimenti interni e così via. Vi sono, cioè, operai, che, dalla mattina alla sera, non fanno altro che montare carburatori e, forse, non hanno la minima idea di come si installi un ammortizzatore; analogamente può capitare che l'operaio addetto alla sistemazione dei sedili non sappia nemmeno il numero dei cilindri del motore presente sulla stessa auto.

Se, però, consideriamo che per fabbricare 12000 auto è necessario il lavoro organizzato di 1000 operai, possiamo far finta di ritenere che ogni operaio abbia realizzato 12 auto, pur se ciò non corrisponde, come abbiamo visto, alla realtà. Tuttavia il valore che ne risulta può essere utilissimo per effettuare confronti con la produttività di altri Paesi.

Seguiremo, ora, un ragionamento che può sembrare l'esatto contrario di quanto affermato poco fa! Per realizzare tutte le funzioni richieste da un computer, non è necessario avere un micro per ciascuna funzione, ma è sufficiente installarvi un unico "cervello" che ciclicamente, in brevissimi istanti successivi, compia numerose routine altrettanto brevi.

In pratica, nell'arco di un periodo brevissimo (anche dell'ordine di centesimi di secondo) il micro sovrintende al video; l'istante successivo lo stesso micro aggiorna l'orologio; quello dopo forse lo dedica allo spostamento dello sprite; quindi provvede, nell'istante successivo, a considerare eventuali mu-

tamenti avvenuti nel corso della scansione della tastiera; poi, magari, del joystick; e così via, di controllo in controllo, di verifica in verifica, fino ad espletare l'ultimo comando. Subito dopo ricomincia daccapo, all'infinito.

Si tratta, quindi, di eseguire migliaia di compiti, ma non tutti insieme, bensì frazionati nell'arco di tempo a disposizione, in modo tale che i mille compiti sembrano compiuti contemporaneamente. Allo stesso modo, nella fabbrica di automobili prima considerata, sembra che ogni operaio produca un'auto e tale impressione, come abbiamo già detto, è utile per gli scopi prefissati.

Quando noi giochiamo con un videogame, insomma, non ci interessa sapere se un particolare effetto grafico o sonoro è compiuto in più fasi successive, pur se molto ravvicinate fra loro: in quel momento interessa giocare e diveritirsi e non elucubrare sulla tacnica adottata per la frammentazione dei compiti.

La tecnica della partizione di tempo (time sharing) fu realizzata sfruttando la capacità di un microprocessore ad operare ad altissima velocità. Molti utenti, quindi, potevano lavorare contemporaneamente mediante altrettanti terminali collegati ad un unico calcolatore. Questo elaborava ciclicamente i vari programmi in modo tale che il singolo utente aveva l'impressione che il computer lavorasse solo per lui.

Naturalmente vi sono dei limiti che non possono esser superati, pena la riduzione drastica del tempo a disposizione di ogni utilizzatore.

Ricorreremo, ora, ad alcuni esempi pratici che rendano più chiaro il concetto di multiprogrammazione pur disponendo di un piccolo computer e, per giunta, lavorando in Basic.

Sorvoleremo, ovviamente, sulle differenze esistenti tra i numerosi termini (time sharing, multiprogrammazione, multi tasking) che, nella comune accezione, indicano procedure piuttosto diverse tra loro.

I programmi di esempio

Il primo programma è incredibilmente semplice e banale e servirà solo per puntualizzare alcuni concetti già esposti in precedenza:

```
100 REM PRIMO PROGRAMMA
110 :
120 PRINT CHR$(147):REM CANCELLA SCHERMO
130 PRINT CHR$(19)"SONO LE ORE:"TI$
140 GOTO 130: REM CHIUSURA CICLO
```

Se date il Run vedrete che lo schermo sarà cancellato (riga 120) e, subito dopo, verrà costantemente (riga 140) visualizzata l'ora, preceduta da un breve messaggio (riga 130).

Naturalmente l'orario sarà quello disponibile al momento; volendo regolare l'orologio sarà sufficiente aggiungere la seguente riga...

```
90 INPUT"DIGITA L'ORA";TI$
```

...ricordandosi di rispondere, alla domanda, con le sei cifre relative, rispettivamente, ad ore, minuti e secondi.

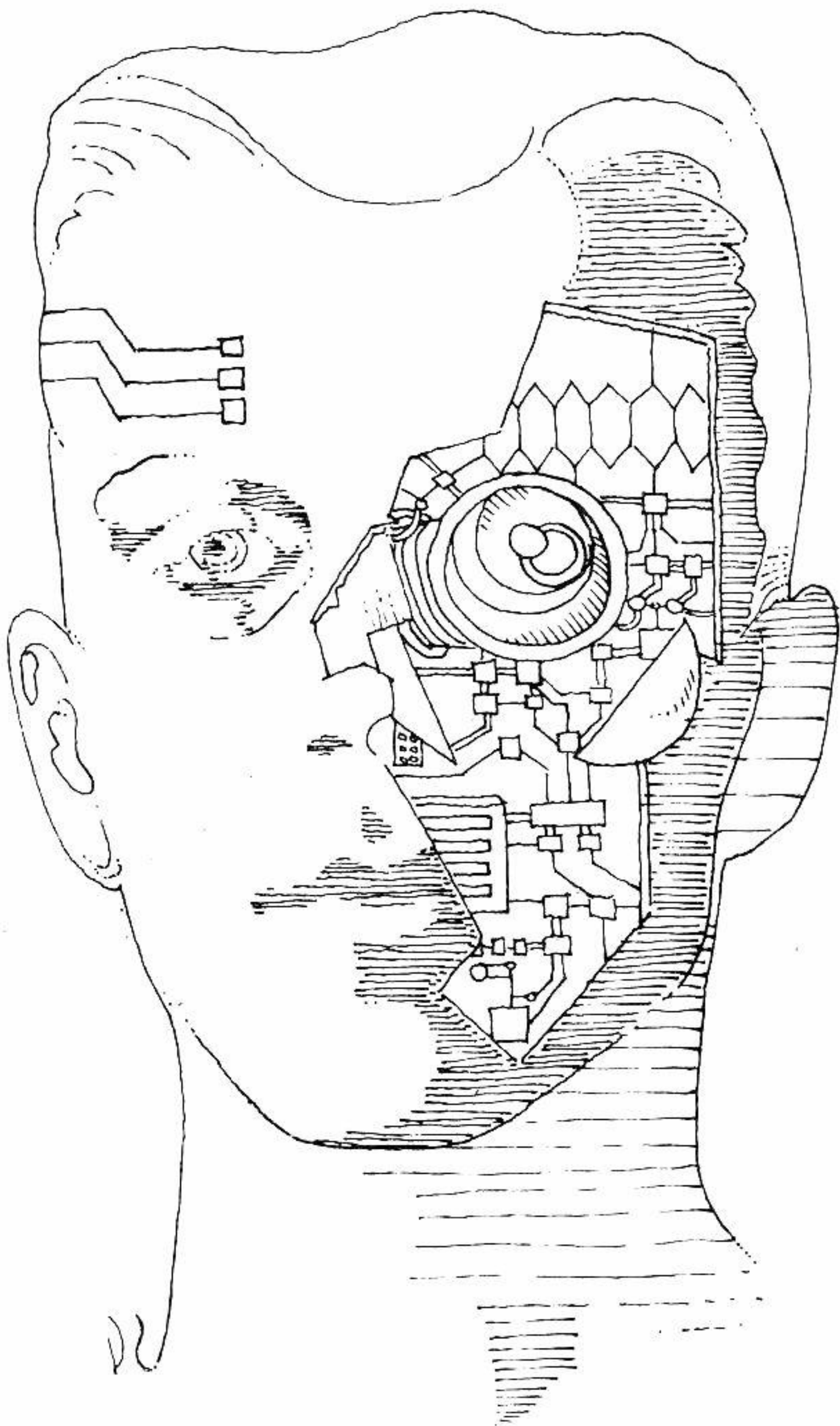
Ma ciò che interessava sottolineare a proposito del semplicissimo programma, era l'individuazione dello svolgimento contemporaneo di più programmi.

Anzitutto, ovviamente, è in azione il programma Basic che, grazie al Goto di riga 140, permette di funzionare all'infinito. E' attivo, inoltre, anche il programma che aggiorna di continuo l'orologio interno del computer, e lo dimostra la variazione continua dell'orario visualizzato; tale azione di aggiornamento, non prevista esplicitamente dal programma Basic, è assolta in modo "trasparente" dallo stesso computer, vale a dire, appunto, che il programmatore non deve preoccuparsi di creare una routine apposita, ma è sufficiente limitarsi a far visualizzare la variabile TI$.

Oltre a quelle viste è però attiva, in posizione di "all'erta", anche la routine che esamina se per caso non sia stato premuto il tasto Run/Stop; lo dimostra il fatto che, se lo premiamo, il programma si arresta.

Tralasciamo, ovviamente, di considerare le numerose altre routine che sovrintendono alla gestione del video ed alle altre complesse procedure vitali per il corretto funzionamento del calcolatore.

Il programma Basic, però, si limita a visualizzare costantemente l'ora e, anche se fossero presenti altre istruzioni "dopo" la riga 140, non potrebbero essere elaborate a causa del Goto 130.

Esiste un modo per far elaborare altri programmi?

Due programmi in time sharing

Supponiamo, ora, di voler determinare il numero di volte in cui è stato premuto un certo tasto, ad esempio il pasto più (+). Pur se tale programma non risulta particolarmente utile, consideriamo una sua possibile versione che non commentiamo a causa della sua enorme semplicità.

```
100 REM SECONDO PROGRAMMA
110 :
120 PRINT CHR$(147):REM CANCELLA SCHERMO
130 GET A$:REM ESAMINA TASTO PREMUTO
140 :
150 IF A$="+" THEN A=A+1
160 PRINT CHR$(19)
170 PRINT "HAI PREMUTO IL TASTO +"
180 PRINT A "  VOLTE"
190 GOTO 130
```

Questo listato assomiglia moltissimo a quello visto in precedenza perchè (riga 190) continua all'infinito. L'artefice che consente un giro non-stop è l'istruzione Get (riga 130) che, se interpellata, assegna alla variabile designata (A$) il tasto premuto (o la stringa nulla in caso negativo) e prosegue nell'elaborazione.

Sfrutteremo proprio l'istruzione Get per fondere i due programmi studiati finora (e, visto che ci siamo, anche altri due) per realizzare l'illusione dell'elaborazione contemporanea di più programmi.

Digitate il seguente listato, che commenteremo subito dopo:

```
100 REM QUATTRO PROGRAMMI
110 REM IN TIME-SHARING
120 REM (QUALSIASI cOMMODORE)
130 :
140 PRINT CHR$(147):REM CANCELLA SCHERMO
150 FOR X=1 TO 8:X$=X$+CHR$(32):NEXT
160 X$=CHR$(157)+X$
170 GET A$:REM ESAMINA TASTO PREMUTO
180 :
190 IF A$="+" THEN A=A+1:PI=PI+1:REM INIZIO PRO-
```

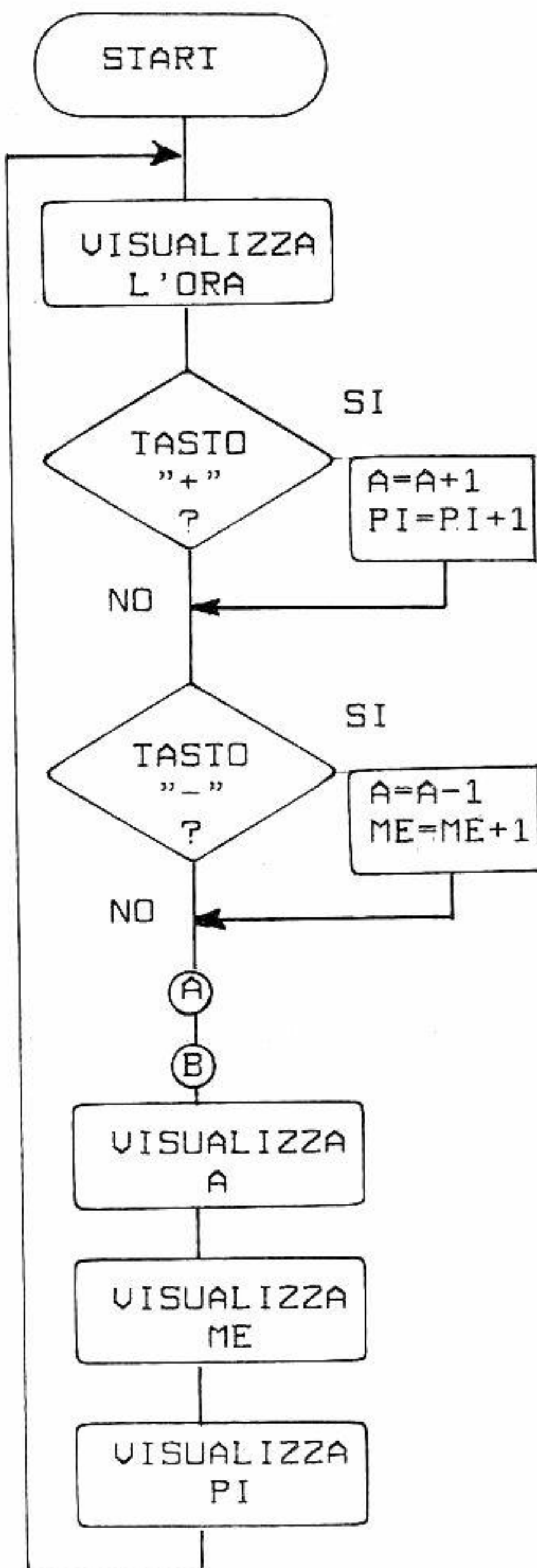
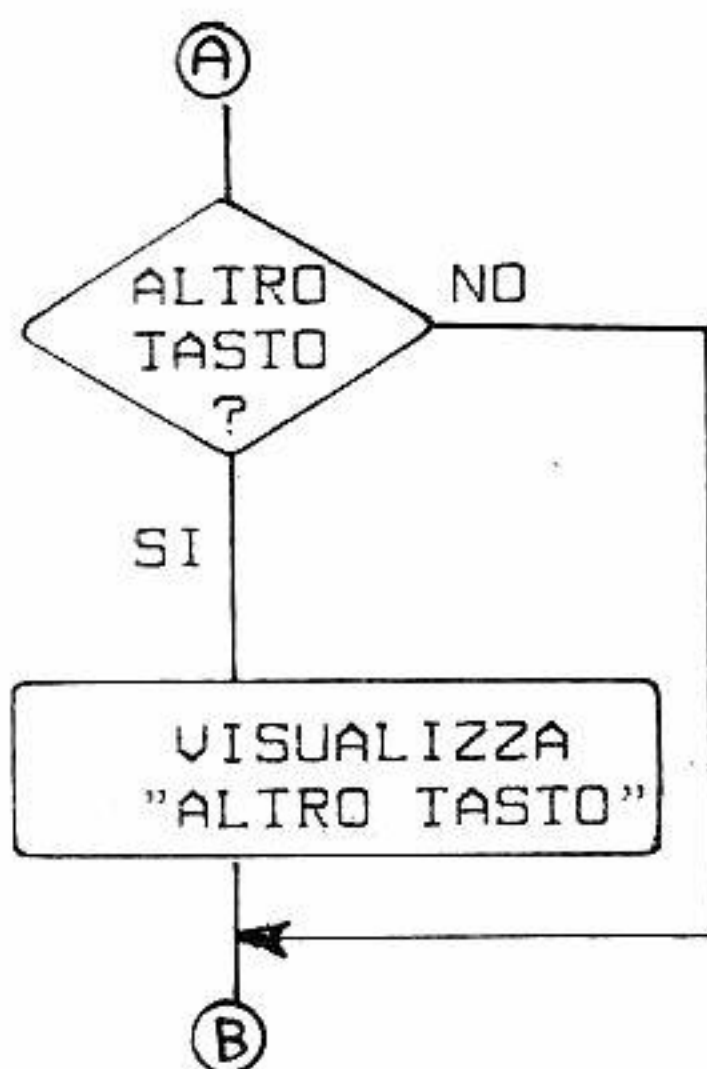
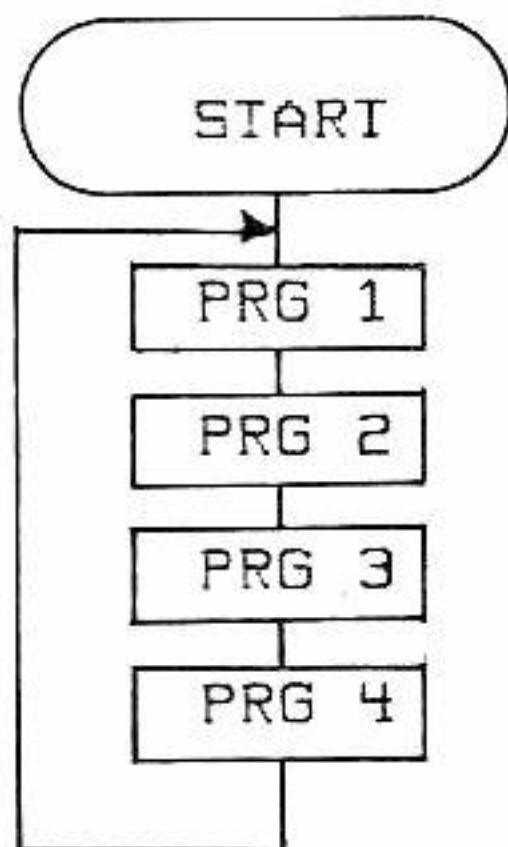
```
GRAMMI N.1 E N. 2
200 IF A$="-" THEN A=A-1:ME=ME+1:REM ANCORA
"PROGRAMMA" N.1 + PRG. N.3
210 PRINT CHR$(19)"SONO LE ORE:" TI$:REM QUARTO
PROGRAMMA
220 PRINT: PRINT "IL VALORE ATTUALE DI A E':"A;X$: REM
PARTE DI PRG N.1
230 PRINT: PRINT "HAI PREMUTO IL TASTO +"PI" VOLTE":
REM ULTIMA PARTE PRG. N.2
240 PRINT "HAI PREMUTO IL TASTO -" ME " VOLTE": REM
ULTIMA PARTE PRG. N.3
250 :
260 GOTO 170:REM CHIUSURA CICLO
```

Anche ad un esame superficiale ci si può accorgere che i programmi attivi stavolta sono ben quattro, pur se frammentati all'interno di un unico, lungo listato. Trascurando la parte iniziale (righe 100-160) destinata ad inizializzare alcune variabili, possiamo approfondire lo studio dei quattro listati:

- Il primo (parte delle righe 190 e 220) si incarica di determinare il numero di volte in cui un certo tasto (ed esattamente il tasto "+") sia stato premuto.
- Il secondo (righe 200 e 240) determina il numero di volte in cui il tasto meno ("-") è stato premuto.
- Il terzo si incarica di effettuare, e visualizzare, la somma algebrica del numero di volte in cui sono stati premuti i due tasti: ad ogni pressione del (+) la variabile "A" viene incrementata; il contrario avviene tutte le volte che si preme (-).
- Il quarto programma si incarica (riga 210) di visualizzare costantemente l'ora.

Impartendo il consueto Run, si potrà notare che, apparentemente, solo il programma relativo alla visualizzazione dell'ora è attivo, mentre gli altri sembrano non entrare in gioco.

Non appena, però, si preme uno dei tasti (+) oppure (-) ci si accorge, grazie alla immediata variazione dei valori visualizzati, che i programmi erano lì, in agguato, pronti a funzionare se "chiamati" opportunamente. Anche premendo i due tasti (+) e (-) con una certa rapidità i risultati ci inducono ad immaginare, con un minimo di fantasia, che i quattro programmi funzionino contemporaneamente.

START
PRG 1
PRG 2
PRG 3
PRG 4
A
ALTRO
TASTO
?
NO
SI
VISUALIZZA
"ALTRO TASTO"
B
START
VISUALIZZA
L'ORA
TASTO
"+"
?
SI
A=A+1
PI=PI+1
NO
TASTO
"–"
?
SI
A=A-1
ME=ME+1
NO
A
B
VISUALIZZA
A
VISUALIZZA
ME
VISUALIZZA
PI

Esaminando il diagramma di flusso relativo, non possiamo fare a meno di notare che il ciclo, prima o poi, si richiude su se stesso ed intuiamo che, esagerando nell'allungare il ciclo (inserendo altre elaborazioni) i due istanti successivi relativi ad uno stesso programma vengono distanziati eccessivamente fra loro; in tal caso non sarà più possibile "illudersi" di veder girare contemporaneamente i numerosi programmi e la rapidità di risposta diventerebbe intollerabile.

Provate, infatti, ad aggiungere queste due righe al programma di prima...

```
251 IF A$<>"+" AND A$<>"-" AND A$<>"" THEN PRINT"HAI
PREMUTO UN ALTRO TASTO"
252 PRINT"........................."
```

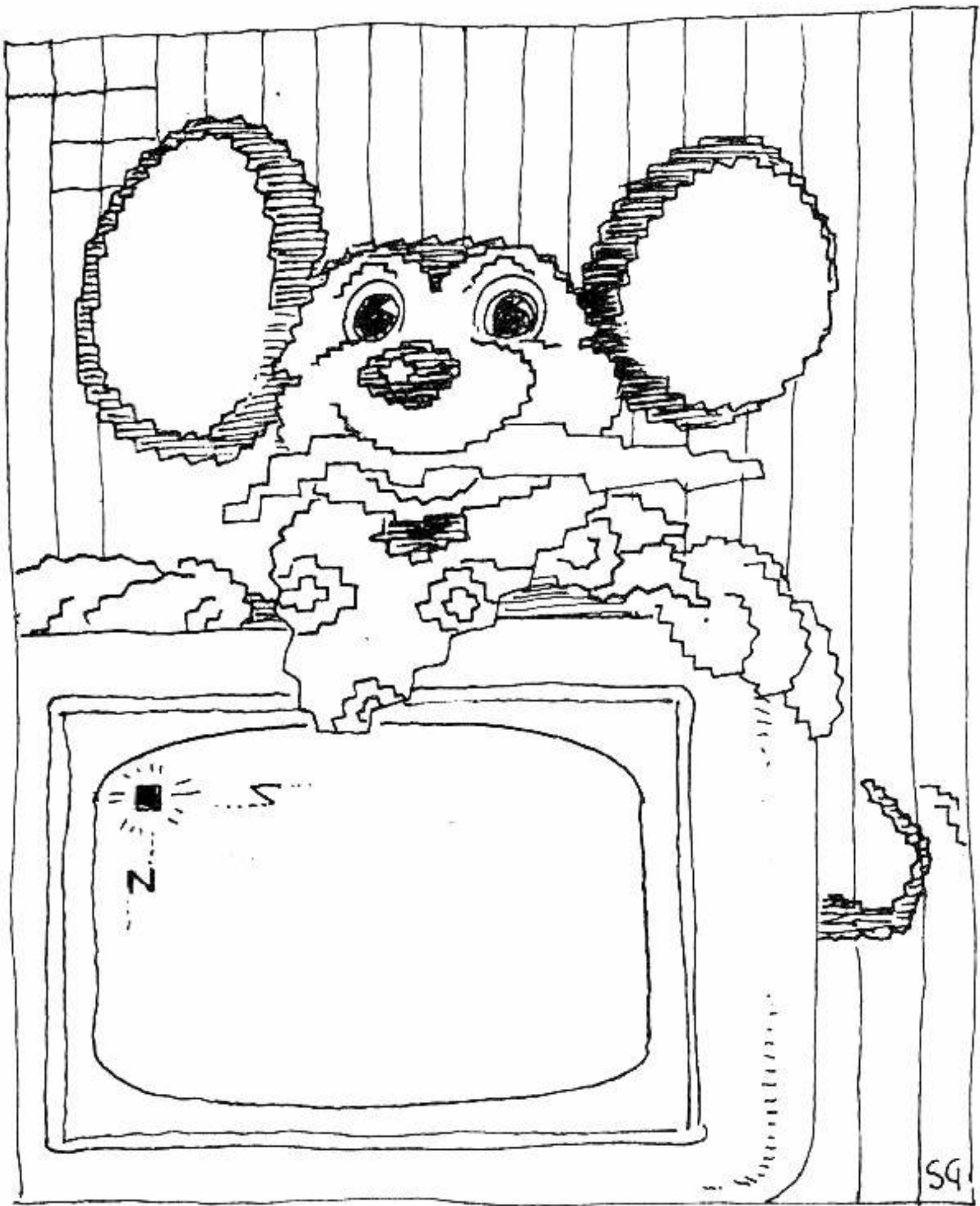
...che hanno il compito di accorgersi se sono stati premuti tasti diversi da (+) e (-). Impartendo il Run, e premendo piuttosto rapidamente molti tasti (tra cui, magari, anche (+) e (-), vi accorgerete che, dopo aver lasciato la tastiera, il programma non può più nascondere il tempo eccessivo che viene richiesto nell'elaborazione dei cinque programmi "contemporanei".

Abbiamo visto, quindi, che con un modesto home computer (che lavora, per giunta, in Basic) è piuttosto difficile organizzare un lavoro secondo una procedura che richiede la partizione di tempo (ma sarebbe più opportuno parlare, nel caso particolare, di partizione "di spazio", più che di "tempo").

Un altro limite notevole che impedisce il ricorso intensivo all'istruzione Get, è sicuramente rappresentato dal fatto che, con tale istruzione, è impossibile, o quantomeno problematico, trasferire al computer un intero messaggio, oppure un numero, che richiedano la digitazione di più caratteri in successione.

L'istruzione idonea a tale scopo è infatti Input che, però, al momento della sua attivazione, interrompe il programma Basic che viene ripreso solo alla pressione del tasto Return.

Per dimostrare la difficoltà del caso si consideri il seguente programma...

```
100 REM DUE PROGRAMMI DI CUI UNO
110 REM CONTENENTE ISTRUZIONI INPUT
120 :
```

```
130 PRINT CHR$(147):REM CANCELLA SCHERMO
140 GET A$:REM ESAMINA TASTO PREMUTO
150 IF A$="" THEN GOSUB 220:GOTO 140
160 PRINT CHR$(147): GOSUB 220: PRINT
170 INPUT"MINUENDO";MI
180 INPUT"SOTTRAENDO";SO: PRINT
190 PRINT MI "-" SO "=" MI-SO
200 GOTO 140
210 :
220 PRINTCHR$(19)"SONO LE ORE:"TI$
230 RETURN
```

...che dovrebbe visualizzare costantemente l'ora oltre a consentire il calcolo di una sottrazione (righe 170-190).

Non appena di impartisce il Run, infatti, l'ora viene visualizzata finchè non si preme un tasto. A questo punto verranno richiesti il minuendo ed il sottraendo ma, finchè non si preme per la seconda volta il tasto Return, il tempo non verrà più scandito istante per istante. E ovvio che, non appena si preme il secondo Return, verrà visualizzata l'ora aggiornata, dal momento che, anche durante la fase di Input, il vostro computer ha provveduto ad aggiornarla in modo "trasparente". Viene però persa la caratteristica di contemporaneità e di tempo reale che contraddistingueva i programmi studiati in precedenza.

Lavorando in Basic, pertanto, la tecnica di pseudo-time sharing può essere adottata con successo solo durante la visualizzazione di un menu, quando, cioè, il programma principale attende l'indicazione di una preferenza. Nel caso non venga premuto alcun tasto è possibile approfittare per visualizzare l'ora, per cambiare continuamente i colori del bordo, per emettere dei segnali sonori e così via.

L'Interrupt

Esiste una tecnica particolare che viene spesso usata dai programmatori per fare in modo che una certa routine venga usata continuamente durante l'uso normale del computer.

Tale tecnica approfitta del fatto che, ogni sessantesimo di secondo, il microprocessore interrompe il lavoro che sta eseguendo e salta ad eseguire alcuni controlli, verifiche ed altre "cose" necessarie per mandare avanti la baracca.

Naturalmente la procedura di Interrupt è trasparente per l'utente finale che non si accorge minimamente delle frequentissime interruzioni.

Ogni sessantesimo di secondo (caso del C/64) il micro salta ad eseguire una routine in linguaggio macchina il cui indirizzo di partenza (nella consueta forma basso/alto) è memorizzato nelle due locazioni adiacenti 788 e 789. Di solito, non appena si accende il C/64, in tali locazioni sono presenti, rispettivamente, i valori 49 e 234. Questi, combinati nella forma 234*256+49, forniscono il valore 59953 che, appunto, rappresenta l'indirizzo di partenza della routine in linguaggio macchina richiamata ogni sessantesimo di secondo.

Poichè i due puntatori 788 e 789 sono in una zona Ram, risulta relativamente semplice, per l'utente finale, modificarne il contenuto e fare in modo che una sua personale routine sia "attraversata" prima di compiere il solito giro di perlustrazione allo scattare del sessantesimo di secondo.

C'è solo da tener presente che la routine da inserire nel ciclo di Interrupt deve assolutamente terminare con l'istruzione di salto incondizionato JMP EA31; tale accorgimento è indispensabile perchè, appunto, il ciclo di interrupt non può fare a meno di essere eseguito.

Si ricorda che EA31, tradotto in decimale, rappresenta proprio il numero 59953 prima visto.

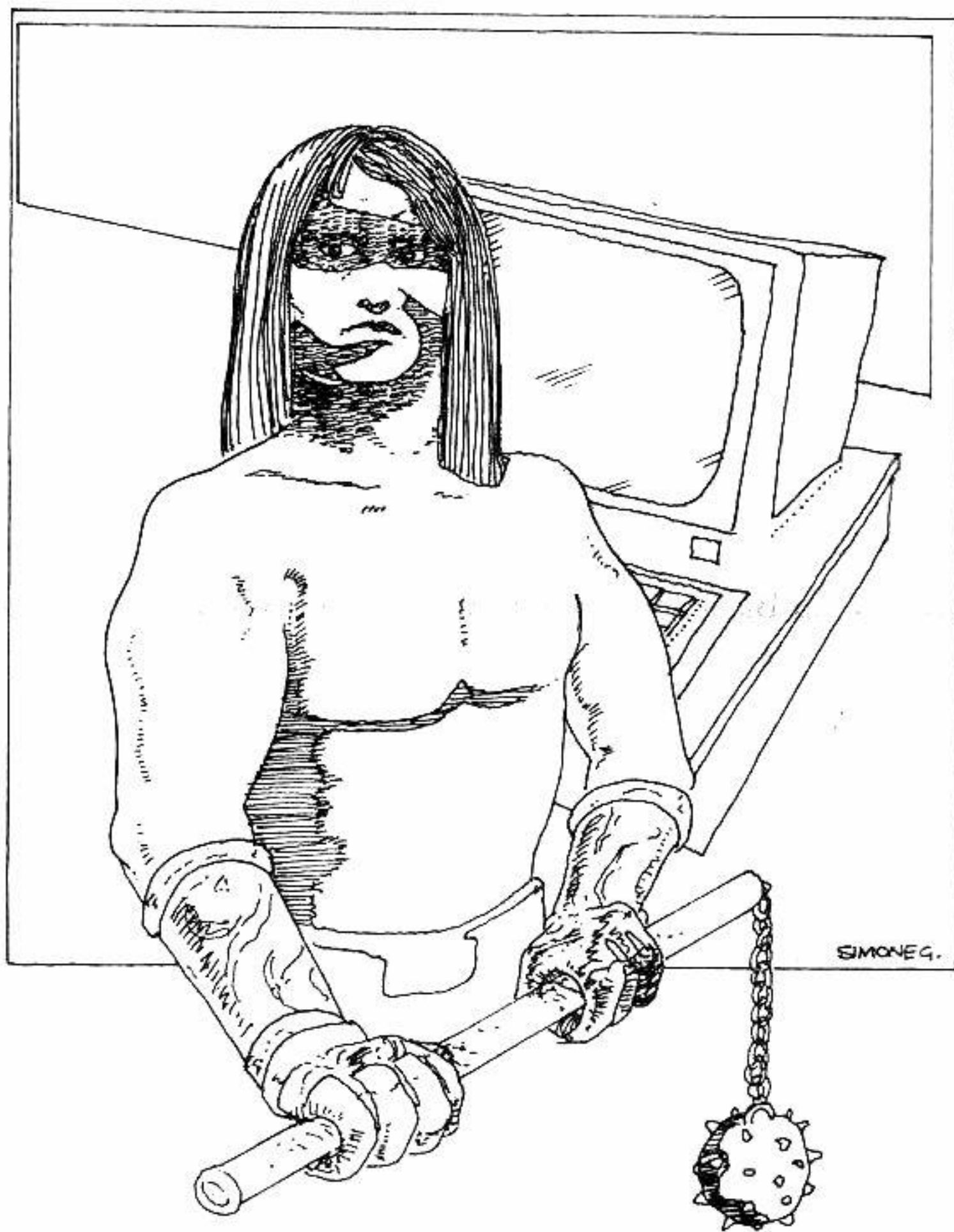
Poichè la routine da inserire deve, necessariamente, essere scritta in linguaggio macchina, verrà il momento in cui si dovrà alterare il puntatore 788/789. Se, però, eseguiamo la modifica, da Basic, per mezzo di istruzioni Poke, corriamo il rischio che, subito dopo aver eseguito la Poke relativa alla locazione 788, arrivi l'impulso di Interrupt che andrebbe ad eseguire la routine allocata in un indirizzo modificato... a metà!

Per evitare questo inconveniente vi sono due strade: la prima, complicata, richiede l'uso delle istruzioni L.M. che riguardano l'Interrupt (CLI, SEI) che costringono il microprocessore a trascurare, oppure ad accettare, le richieste di interruzioni.

Il secondo metodo, decisamente più semplice, consiste nell'allocare la nostra routine L.M. in una locazione che abbia, come parte bassa dell'indirizzo nuovo, la stessa parte bassa (49) dell'indirizzo originario. In tal modo modificando, anche da Basic mediante Poke, la sola locazione 789, si renderà disponibile un efficace deviatore, governabile da Basic, che

renderà attiva, o meno, la nostra routine.

In altre parole, se le due locazioni contengono normalmente 49 e 234, faremo in modo che la nostra routine sia allocata a partire da 49201, indirizzo fornito dai due valori 49 e 192 opportunamente combinati tra loro (192*256+49).

Con tale semplicissimo accorgimento sarebbe possibile, ma solo in teoria, selezionare ben 256 cicli diversi di Interrupt limitandosi a modificare, mediante Poke, la locazione 789 e a patto, ovviamente, di scrivere altrettante routine nelle 256 pagine di cui dispone il microprocessore. Ad approfondire tale discorso, però, potrebbero essere disponibili solo i lettori più esperti ed è questo il motivo per cui non andiamo oltre.

Dal semplice al difficile

Provate, ora, a digitare il seguente listato, e fatelo partire con un semplice Run.

```
100 PRINT CHR$(147)
105 A$=CHR$(32) + CHR$(32)
110 PRINT CHR$(19): FOR I=160 TO 162
120 PRINT A$ STR$(PEEK(I))A$;: NEXT
130 PRINT: GOTO 110
```

Rintracciare il motivo per cui si utilizzano tante istruzioni apparentemente superflue sarà un valido esercizio per il lettore che voglia approfondire la gestione della visualizzazione di numerosi valori numerici su di uno stesso rigo.

Ciò che vogliamo mettere in evidenza, invece, è solo il fatto che in un C/64 vi sono tre locazioni di memoria (160, 161 e 162) che sono interessate alla manipolazione del... tempo.

Tutti dovrebbero sapere, inoltre, che è possibile visualizzare un carattere sullo schermo mediante istruzioni Poke che interessano l'area del video, allocata da 1024 a 2023 (in esadecimale da 0400 a 07E7).

Un programma analogo a quello appena visto, che ricorra a Poke anzichè a Print, può essere il seguente:

```
100 PRINT CHR$(147)
110 FOR I=0 TO 2
120 POKE 1024+I, PEEK(160+I)
130 NEXT: GOTO 110
```

Naturalmente quest'ultimo può presentare difficoltà di comprensione rispetto al precedente perchè i tre caratteri visualizzati nelle prime tre celle video cambiano continuamente e sembrano non avere alcun senso.

Quest'ultimo programma, però, ha il vantaggio di essere facilmente inseribile all'interno di una routine di Interrupt mediante, ad esempio, il seguente programma Basic che contiene, al suo interno, informazioni sufficienti per la comprensione del programma in linguaggio macchina:

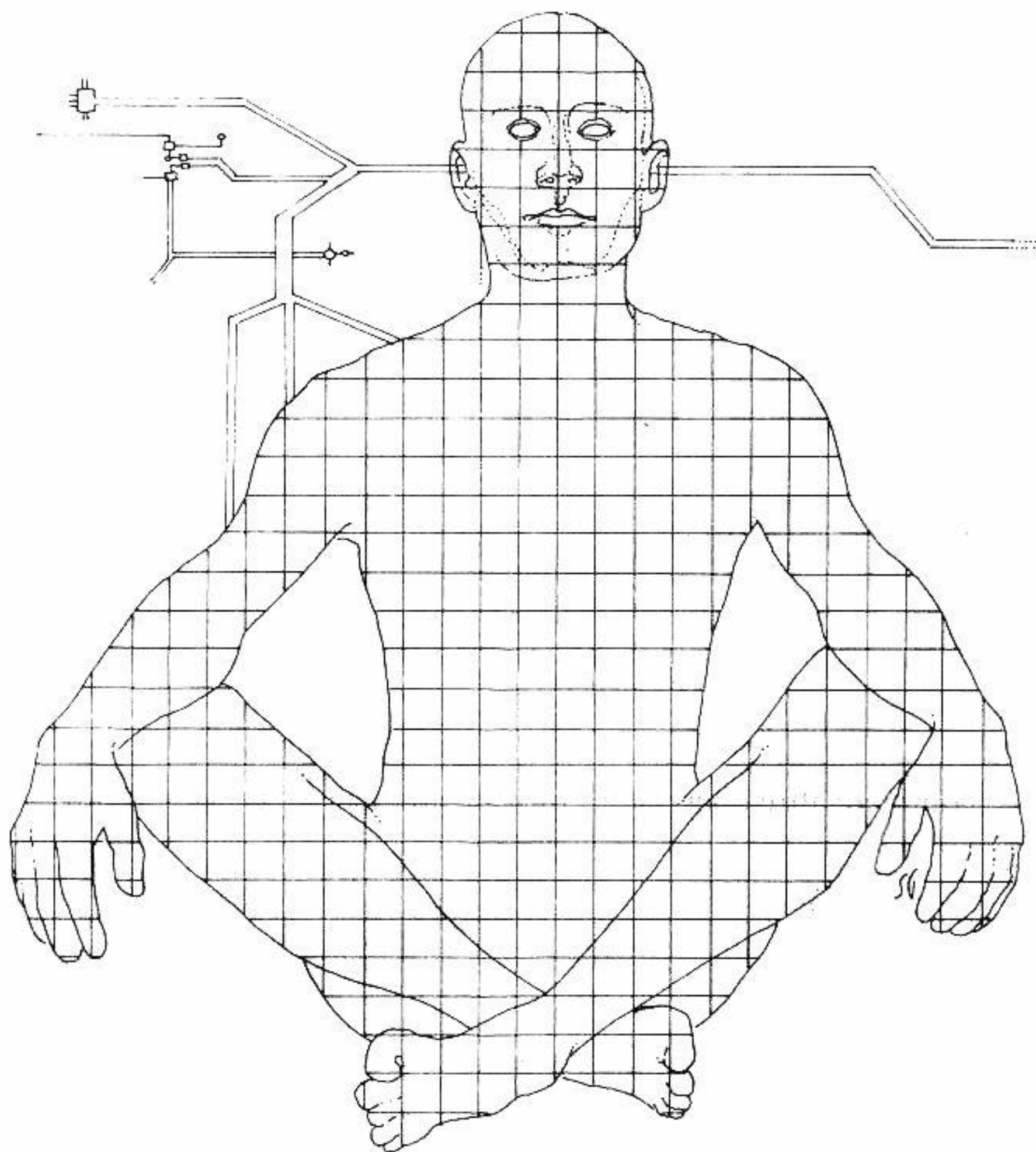
SG

```
100 REM OROLOGIO NELL'INTERRUPT
110 REM LE TRE LOCAZIONI
120 :
130 READ X: IF X=-1 THEN 150
140 POKE 49201+I,X: I=I+1: GOTO130
150 POKE 789,192: REM CAMBIA INTERRUPT
160 :
170 DATA 162,3    :REM LDX#3
180 DATA 181,159  :REM LDA,X #159
190 DATA 157,255,3:REM STA,X #1023
200 DATA 202      :REM DEX
210 DATA 208,248  :REM BNE F8
220 DATA 76,49,234:REM JMP #59953
230 :
240 DATA -1:REM FINE DEI DATA
```

Sorvolando il commento sulle istruzioni L.M. (che richiederebbe un intero fascicolo) ci limitiamo ad evidenziare l'unica Poke necessaria per il dirottamento della routine di Interrupt e la riga 220 contenente i codici relativi al salto incondizionato JMPEA31 di cui abbiamo parlato in precedenza.

Non appena fate partire il programma con un semplice Run vi accorgete che nelle prime tre locazioni dello schermo vengono costantemente visualizzati i tre caratteri che rappresentano il contenuto delle tre locazioni 160, 161 e 162. Ora, però, possiamo utilizzare il computer per far girare altri programmi o per compiere altre operazioni: provate ad eseguire un List, a cancellare righe, a scrivere un nuovo programma: durante tutte queste operazioni le prime tre celle video continueranno indisturbate a visualizzare, a modo loro, l'ora in tempo reale.

Con l'istruzione Poke 789,234 (oppure con la pressione di Run/Stop e Restore) si ripristinano i valori originali.
Abbiamo con ciò dimostrato come sia semplice(!) obbligare un qualsiasi computer a compiere più procedure in modo apparentemente contemporaneo.

Nei primi esempi abbiamo visto come, con il Basic, si manifestino subito dei limiti inaccettabili. Possiamo ora affermare che, ricorrendo al Linguaggio Macchina (ed in particolare alla manipolazione del vettore di Interrupt) è possibile risolvere il problema una volta per tutte?

Il caso si complica

Vediamo che cosa succede se, anzichè visualizzare solo tre locazioni, proviamo a visualizzarne ben 256; un programma per il C/64 può essere il seguente, che consigliamo di trascrivere dopo aver premuto i tasti Run/Stop e Restore:

```
100 REM VISUALIZZA 256 BYTE IN INTERRUPT
110 REM DOPPIO CICLO DI RITARDO
120 :
130 PRINT CHR$(147) "CONTO FINO A 10000 IN...";: GOSUB
340: PRINT" SECONDI"
140 PRINT: PRINT "PREMI UN TASTO PER CON-
TINUARE"
150 GET A$: IF A$="" THEN 150
160 :
170 READ X: IF X=-1 THEN 190
180 POKE 49201+I,X: I=I+1: GOTO 170
190 POKE 789,192: REM CAMBIA INTERRUPT
200 :
220 DATA 162,255  :REM LDX#255
230 DATA 181,0    :REM LDA,X $0
240 DATA 157,255,3:REM STA,X #1023
250 DATA 202      :REM DEX
260 DATA 208,248  :REM BNE F8
290 DATA 76,49,234:REM JMP #59953
300 :
310 DATA -1:REM FINE DEI DATA
320 FOR I=1 TO 10: PRINT: NEXT
330 PRINT"MENTRE ORA IMPIEGO...";:GOSUB340:END
340 TI$="000000":FOR A=1 TO 10000: NEXT: PRINT TI$;:
RETURN
```

Poco dopo il Run la parte superiore dello schermo, ed esattamente le prime sei righe per intero e i primi 14 byte del settimo rigo (per un totale di 256 caratteri) si riempono di caratteri apparentemente casuali. In effetti rappresenta la radiografia, in tempo reale, della prima "pagina" di memoria in cui, oltre alle già note locazioni 160, 161 e 162, vi sono altre locazioni che sovrintendono a particolari funzioni: provate, quando ricompare il familiare READY, a premere e rilasciare il tasto Control, oppure a far salire e scendere il cursore o a premere Run/Stop:

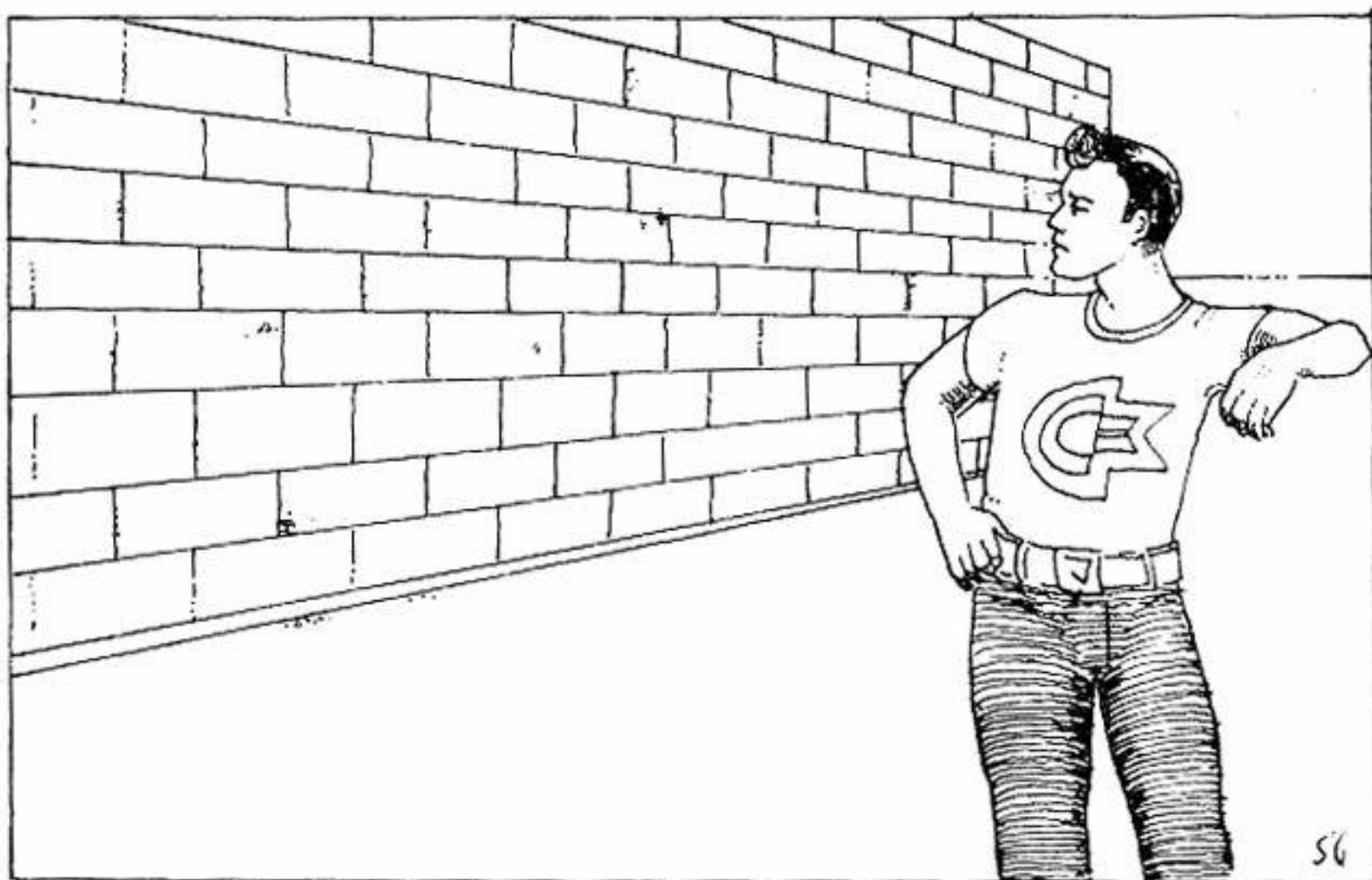
se avete pazienza (ed una mappa di memoria), potete individuare la locazione interessata alle singole operazioni accenate.

Ma ciò che volevamo mettere in evidenza non era tanto un sistema per radiografare una pagina di memoria ma, soprattutto, il fatto che anche lavorando in linguaggio macchina, notoriamente iperveloce, si accusano alcuni rallentamenti che possono diventare intollerabili.

A parte, infatti, il tempo nettamente maggiore che il computer impiega per contare da 1 a 10000, ci accorgiamo che alcune semplici operazioni vengono svolte in tempi maggiori: provate a dare un comando di List e ve ne accorgerete.

Se poi, dopo avere premuto Run/Stop e Restore, aggiungete le seguenti linee...

```
210 DATA 160,4 :REM LDY#4
270 DATA 136 :REM DEY
280 DATA 208,243 :REM BNE F3
```

...che, facendo intervenire il registro Y, costringono a visualizzare quattro volte la pagina zero, vi accorgerete di un incredibile rallentamento delle operazioni, avvertibile perfino nello spostare il cursore: nella fase di List, addirittura, è possibile notare la particolare tecnica seguita dal computer per la realizzazione dello scrolling.

Che fare?

Da quanto detto possiamo quindi concludere che la possibilità di far funzionare più programmi in contemporanea si scontra, prima o poi, con le caratteristiche peculiari del sistema adoperato.

Pur ricorrendo a tecniche di programmazioni pregevoli, in grado di risparmiare fino all'ultimo byte, e metodologie sofisticate, che sfruttano al massimo le risorse del microprocessore, non è pensabile andare oltre un certo limite, superato il quale si può addirittura provocare il crash del sistema.

Ecco quindi il motivo per cui non è importante avere a disposizione vaste quantità di memoria o disporre di linguaggi in grado di svolgere funzioni incredibilmente complesse; è fondamentale, soprattutto, individuare la massima velocità raggiungibile dal micro (in termini di megahertz), la sofisticazione delle istruzioni di cui dispone, il numero di cicli macchina che richiede per svolgerle e, in ultima analisi della quantità di tecnologia che contiene.

Il caso dell'Amiga

Non potevamo certo fare a meno di ricorrere ad un confronto illuminante che meglio di ogni altro chiarisse che cosa si intenda con il termine "multiprogrammazione" (e analoghi).

L'Amiga è un computer la cui caratteristica peculiare è, appunto, la capacità di multiprogrammazione.

Anche chi non possiede un Amiga avrà sicuramente avuto occasione di vederne in funzione un esemplare, almeno una volta. A coloro che, sfortunati, non appartengono a tale schiera, ricorderò semplicemente che tra i vari file forniti in omaggio con l'Amiga è presente un programma dal nome chiarificatore "Clock" (orologio) che, se attivato, consente di tener visualizzato costantemente un orologio digitale (oppure analogico) con tanto di lancette relative alle ore, ai minuti e ai secondi; è addirittura possibile spostare la finestra dell'orologio in qualsiasi zona dello schermo con il mouse.

Ciò significa che, a parte la possibilità di modificare entro ampi limiti le dimensioni dello stesso orologio (in altezza e larghezza), sarà possibile visualizzare costantemente il quadrante, in tempo reale, anche durante l'esecuzione di altri programmi, scritti in qualsivoglia linguaggio.

Soffermiamoci alcuni istanti per meglio comprendere l'immane lavoro che l'Amiga è chiamata a compiere per riuscire nell'intento: la finestra dell'orologio è in alta risoluzione; ne deriva che, istante per istante, il microprocessore dovrà cancellare la lancetta dei secondi dalla posizione precedente e ricostruirla dopo aver determinato l'angolo, descritto in senso orario, tenendo conto delle dimensioni della finestra e del rapporto tra lunghezza e larghezza che, lo ricordo, possono esser variati a piacere dall'utente in qualsiasi momento con il semplice movimento del mouse.

Quando due lancette si sovrappongono (e ciò capita alla lancetta dei secondi due volte per ciascun giro) bisogna tener conto che, dopo lo spostamento, bisogna far ricomparire anche la lancetta dei minuti (o delle ore, a seconda dei casi).

Con un C/64 tale operazione potrebbe certamente essere svolta ricorrendo alla grafica in alta risoluzione; dubito, però, che l'orologio possa esser gestito con facilità se posizionato all'interno di un ciclo di Interrupt...

Con l'Amiga è possibile caricare più volte il programma Clock e visualizzare contemporaneamente più orologi, di forma e dimensione diverse, che battono all'unisono.

Tanto per rendersi conto della reale multiprogrammazione di cui è capace l'Amiga, tenete presente che, per prova, ho caricato ben 14 (quattordici) orologi, ed ho conferito, ad ognuno di loro, diversa forma e dimensione.

Ebbene è necessario stare molto, ma molto attenti per accorgersi che, in effetti, la lancetta dei secondi non è sistemata nella stessa posizione in un determinato istante, ma si riesce ad individuare uno sfasamento provocato dal fatto che gli orologi sono azionati alternativamente, l'uno dopo l'altro.

Se non è multiprogrammazione questa, sapete indicarmi un computer in grado di raggiungere una rapidità di multiprogrammazione comparabile con quella dell'Amiga?...

AM
PM
SIMONE GALLINA